UNCLASSIFIED

SYSTEMS ENGINEERING
Research Center

# Requirements Management for Net-Centric Enterprises Phase II

## Final Technical Report SERC-2011-TR-017-2

December 31, 2011


Douglas A. Bodner - Georgia Institute of Technology
Nenad Medvidovic - University of Southern California
Jo Ann Lane - University of Southern California
Barry W. Boehm - University of Southern California
William C. Kessler - Georgia Institute of Technology
William B. Rouse - Georgia Institute of Technology
George Edwards - University of Southern California
Kristi Kirkland - Georgia Institute of Technology
Ivo Krka - University of Southern California
Animesh Podar - Georgia Institute of Technology
Daniel Popescu - University of Southern California



Contract Number: H98230-08-D-0171 DO1, TTO2, RT25

**Report No. SERC-2011-TR-017-2**
**December 31, 2011**

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **31 DEC 2011** | | **00-00-2011 to 00-00-2011** |

4. TITLE AND SUBTITLE
**Requirements Management for Net-Centric Enterprises Phase II**

5a. CONTRACT NUMBER
5b. GRANT NUMBER
5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER
5e. TASK NUMBER
5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
**Georgia Institute of Technology,Systems Engineering Research Center,Atlanta,GA,30332**

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT
**Net-centric enterprises increasingly are found in government and industry contexts. In this research, a net-centric enterprise consists of a number of semi-autonomous organizations that collaborate within the context of a federated structure. Such collaborations may be temporary and of known duration, temporary and of unknown duration, or permanent and known to be permanent. When such semi-autonomous organizations collaborate, they typically have information technology needs to support their collaboration. In the information technology (IT) domain, such needs are called requirements. From a business or organizational perspective, these needs are called capabilities or functions. In designing and developing IT systems to support high-level capabilities, capabilities are decomposed to functions and then to requirements. From requirements, software architectures are derived and then implemented. This process occurs in the context of integrating or interoperating systems. The fundamental problem is how to manage the process of proceeding from capabilities to systems, i.e., requirements management in the netcentric enterprise. This is a socio-technical problem involving inter-organizational socio issues, as well as technical system integration issues. This report provides a methodology for addressing the requirements management problem that includes component methods, processes and tools for addressing subproblems. This methodology is evaluated via application to case studies of system integrations that have strong net-centric enterprise characteristics. In addition, case studies are used to elucidate effective practices with respect to socio issues. Validation of the concepts and results of the research is done via interaction with subject matter experts. Finally, recommendations for future research and technology transfer are provided.**

15. SUBJECT TERMS

<table>
<tr><td colspan="3">16. SECURITY CLASSIFICATION OF:</td><td rowspan="2">17. LIMITATION OF ABSTRACT<br>**Same as Report (SAR)**</td><td rowspan="2">18. NUMBER OF PAGES<br>**94**</td><td rowspan="2">19a. NAME OF RESPONSIBLE PERSON</td></tr>
<tr><td>a. REPORT<br>**unclassified**</td><td>b. ABSTRACT<br>**unclassified**</td><td>c. THIS PAGE<br>**unclassified**</td></tr>
</table>

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center (SERC) under Contract H98230-08-D-0171. SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology

# ABSTRACT

Net-centric enterprises increasingly are found in government and industry contexts. In this research, a net-centric enterprise consists of a number of semi-autonomous organizations that collaborate within the context of a federated structure. Such collaborations may be temporary and of known duration, temporary and of unknown duration, or permanent and known to be permanent.

When such semi-autonomous organizations collaborate, they typically have information technology needs to support their collaboration. In the information technology (IT) domain, such needs are called requirements. From a business or organizational perspective, these needs are called capabilities or functions. In designing and developing IT systems to support high-level capabilities, capabilities are decomposed to functions and then to requirements. From requirements, software architectures are derived and then implemented. This process occurs in the context of integrating or interoperating systems. The fundamental problem is how to manage the process of proceeding from capabilities to systems, i.e., requirements management in the net-centric enterprise. This is a socio-technical problem involving inter-organizational socio issues, as well as technical system integration issues.

This report provides a methodology for addressing the requirements management problem that includes component methods, processes and tools for addressing sub-problems. This methodology is evaluated via application to case studies of system integrations that have strong net-centric enterprise characteristics. In addition, case studies are used to elucidate effective practices with respect to socio issues. Validation of the concepts and results of the research is done via interaction with subject matter experts. Finally, recommendations for future research and technology transfer are provided.

This page intentionally left blank

# TABLE OF CONTENTS

# FIGURES AND TABLES

# 1 SUMMARY

This report details the findings of a research effort studying requirements management in net-centric enterprises. In this research, a net-centric enterprise consists of a number of semi-autonomous organizations that collaborate within the context of a federated structure. Such collaborations may be temporary and of known duration, temporary and of unknown duration, or permanent and known to be permanent. When such organizations collaborate, they have IT needs to support the collaboration. Since the organizations have pre-existing IT systems, the collaboration needs are often posed as a system integration or merger.

The existing requirements management capabilities focus mainly on requirements throughout the traditional single-system software development lifecycle. With the increasingly common net-centric enterprise, the management of requirements from enterprise-level capability intents, through the system integration process, thence to integrated system outcomes is also of interest. Of course, this occurs in a multi-stakeholder environment. Hence it is not purely a technical problem, but also has strong socio-technical aspects.

In this report, we specify a taxonomy of integration efforts with the goal of providing pointers to effective requirements management practices. Using a previously developed methodological framework, we elaborate methods, processes and tools within the framework to provide support for capabilities-to-requirements decisions and requirements-to-architectures decisions. These MPTs are analyzed with respect to their effectiveness by applying them to net-centric enterprise case studies that involve systems of systems. We also utilize a case study to analyze socio decision processes and outcomes in a requirements management effort in a net-centric enterprise responsible for design and development of a next-generation fighter plane. Finally, we validate the research concepts and MPTs with surveys and walk-throughs involving subject matter experts in systems integration.

# 2 Introduction

## 2.1 Problem Statement

This research addresses requirements management in the net-centric enterprise. In this context, system integrations and mergers, often of a temporary or unspecified duration, are used to support multi-organizational collaboration. Requirements management then involves identifying, reconciling, documenting, analyzing and prioritizing capabilities, functions and requirements as capabilities are decomposed into requirements and then mapped to architectures. Requirements management should support the traceability of progress on the extent to which capabilities are being realized and should also support the evolution of new capabilities as the net-centric enterprise evolves to support new missions and collaborations.

This research specifies a methodological framework and associated MPTs to enable effective requirements management in a net-centric context.

## 2.2 Objectives

The overall objectives of this research are the following.

- Enable "requirements management" throughout integration lifecycle
  - Requirements definition and reconciliation
  - Traceability
  - Architecture specification
  - Balance between automation and decision support
- Address
  - Organizational differences
  - Selection-from-alternatives vs. design
  - Ambiguity and robustness

## 2.3 Definitions

The following terms are used in this report as defined here.

- <u>Architecture</u> – A variety of definitions for architecture exist. One relevant definition for this research is that an architecture is "the fundamental organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution" (ANSI/IEEE, 2006). This is not meant to be exclusive of other, similar definitions maintained by other standards organizations.

- Capability – Consistent with DoD usage, a capability is the "ability to achieve a desired effect under specified standards and conditions through combinations of ways and means to perform a set of tasks" (CJCS, 2007). More specifically, capability is a high-level business imperative that typically is decomposed into subordinate functions that together achieve the capability.
- Function – A function is an intermediate concept between a capability and a requirement. In a large, complex enterprise, there may be many levels of functions as capabilities are decomposed into functions, thence to requirements.
- Integration – A system integration occurs when two or more systems are joined by developing interfaces between them. Each system, more or less, retains its identity. An integration can be temporary.
- Interoperability – Interoperability is the property of a system whereby the system's interfaces are specifically designed to work with other systems with limited or no interface modifications. Inherent in this definition is that the interfaces are well-understood enough to accommodate working with other systems from an implementation perspective.
- Interoperation – Interoperation occurs when two or more systems are joined together using existing interoperability features of the systems. Interoperation is typically temporary.
- Merger – A system merger occurs when two or more systems are joined by taking the best parts of each and combining them to form a new system. Mergers are typically permanent.
- MPT – Methods, processes and tools are used to solve specific problems in the systems engineering domain.
- Net-centric enterprise – A net-centric enterprise engages a number of semi-autonomous organizations under the umbrella of a federated structure. These organizations have independent but related missions and often collaborate.
- Requirement – In IT systems, a requirement is a particular, well-defined and well-scoped need that must be satisfied by a system. A requirement typically has a stakeholder or stakeholders who advocate for its continued inclusion in the system design and development. In this research, a requirement is generally considered to be at a lower level (i.e., closer to the software design and development process) than a capability or function.
- Requirements definition – Requirements definition, in the traditional software engineering sense, refers to the process of determining requirements for a software system. Here, it also encompasses the determination of high-level capabilities.
- Requirements management – Requirements management is the process whereby capability intents are identified, decomposed into functions, then into requirements for system design, development and evolution. Conflicts and

dependencies between capabilities, functions and requirements must be identified and tracked, with conflicts being resolved. Requirements (including capabilities and functions) must be documented, analyzed, and prioritized. Additionally, requirements management includes the traceability of progress toward meeting higher-level capabilities and functions and lower-level requirements are met. As the system's mission changes, new capabilities, functions and requirements may be added.

- Requirements reconciliation – Requirements reconciliation is a method/process by which different stakeholders in a system are brought together to determine their requirements (or more generally capabilities or functions) for the system, to identify any conflicts, and then negotiate a mutually agreeable solution based on prioritization.
- System-of-systems (SoS) – A system-of-systems is a large-scale, complex system composed of a collection of heterogeneous, independent components that themselves are considered systems. An SoS may be directed, in that it is designed, built and managed for a particular purpose (with the constituent systems normally under control of the overall SoS management), or it may be acknowledged, in that it is centrally managed with the constituent systems retaining their individual autonomy, and with any changes negotiated by the individual systems and the overall SoS management (DoD, 2008).

## 2.3 Presentation of Results

Project results to date were presented at the 2011 Annual SERC Research Review during October 5-6, 2011. Questions and comments from the presentations are compiled in Appendix A, along with responses from the research team as to whether the issues raised represent current work, future work, or work to be done by others (e.g., due to not being research-related work).

## 2.4 Organization of Report

The remainder of this report is organized as follows. Section 3 summarizes the state-of-the-art associated with requirements management in net-centric enterprises, with a focus on integration, merging and inter-operability. Section 4 presents a methodological framework and approach for addressing the problem. A taxonomy of net-centric integration problems is presented in Section 5 with the purpose of pointing toward successful methods of addressing requirements management. Sections 6 and 7 address, respectively, the capabilities-to-requirements engineering and the requirements-to-architectures engineering aspects of our methodology. Given the socio-technical nature of the problem, a case study addressing decision-making in capability and requirements management in net-centric enterprise is presented in Section 8. Section 9 contains validation analysis of the concepts and MPTs specified in this research and identifies critical gaps that need to be addressed in practice. Section 10 reviews existing commercial tools relative to their capabilities and deficiencies regarding these gaps. Finally, Section 11 concludes and provides avenues of future research.

# 3 STATE-OF-THE-ART SUMMARY

System integrations and mergers are increasingly commonplace, as organizations seek to work together to address complex problems. Requirements management in this context is challenging due to multiple stakeholders with perhaps conflicting motivations, changing environmental conditions and changing mission needs.

Traditional requirements management tools address requirements management in the traditional single-system software development process. However, requirements management in the net-centric enterprise, with its emphasis on enterprise level capability intents that cut across multiple stakeholder organizations and systems, remains an area of active research. Our Phase 1 report provides a literature review in this regard (Bodner et al., 2011), and Figure 1 illustrates the unaddressed needs that the net0centric enterprise brings to the requirements management problem.

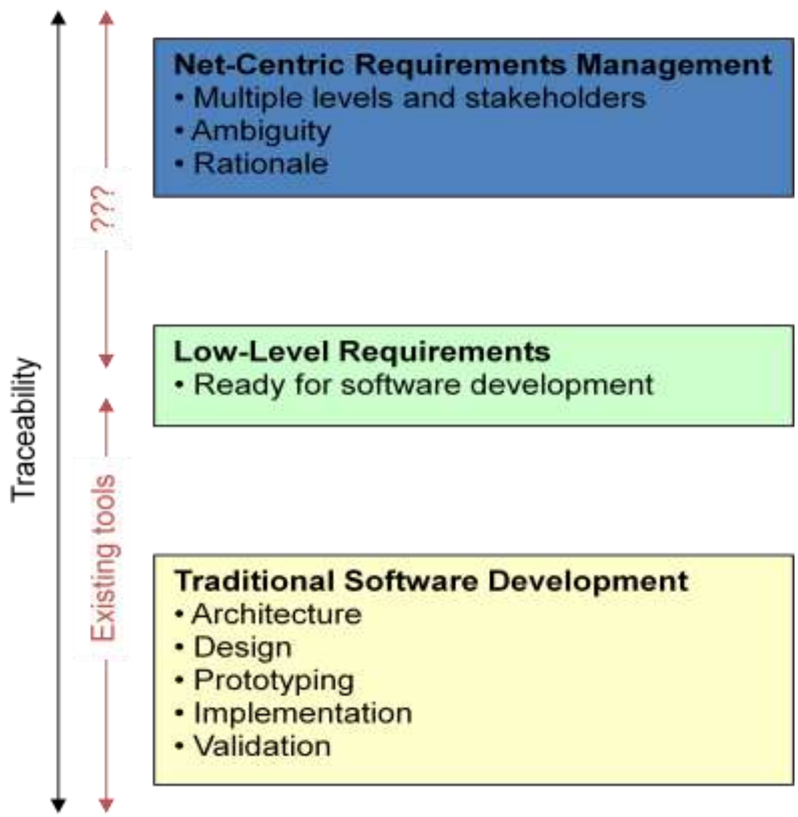


**Figure 1: Requirements Management in a Net-Centric Enterprise**

The lower part of Figure 1 encompasses primarily technical issues associated with software development. Moving up to the net-centric environment, the problem gains much more of a socio-technical flavor, as decision-making, decision authority, trust and negotiation come into play. Research is beginning to address the technical aspects of the integration problems in the net-centric environment (Land & Crnkovic, 2011). However, there is little research that addresses the socio-technical problem. Socio-technical phenomena are increasingly important avenues of research (Bennett, Kessler, & McGinnis, in press). This research addresses the requirements management problem in net-centric enterprises as a socio-technical problem.

# 4 FRAMEWORK AND APPROACH

In Phase 1 of this project, we articulated a methodological framework for addressing the problem of requirements management in a net-centric enterprise. We identified several MPTs that could be adapted to use within this framework. The framework and associated MPTs are shown in Figure 2.

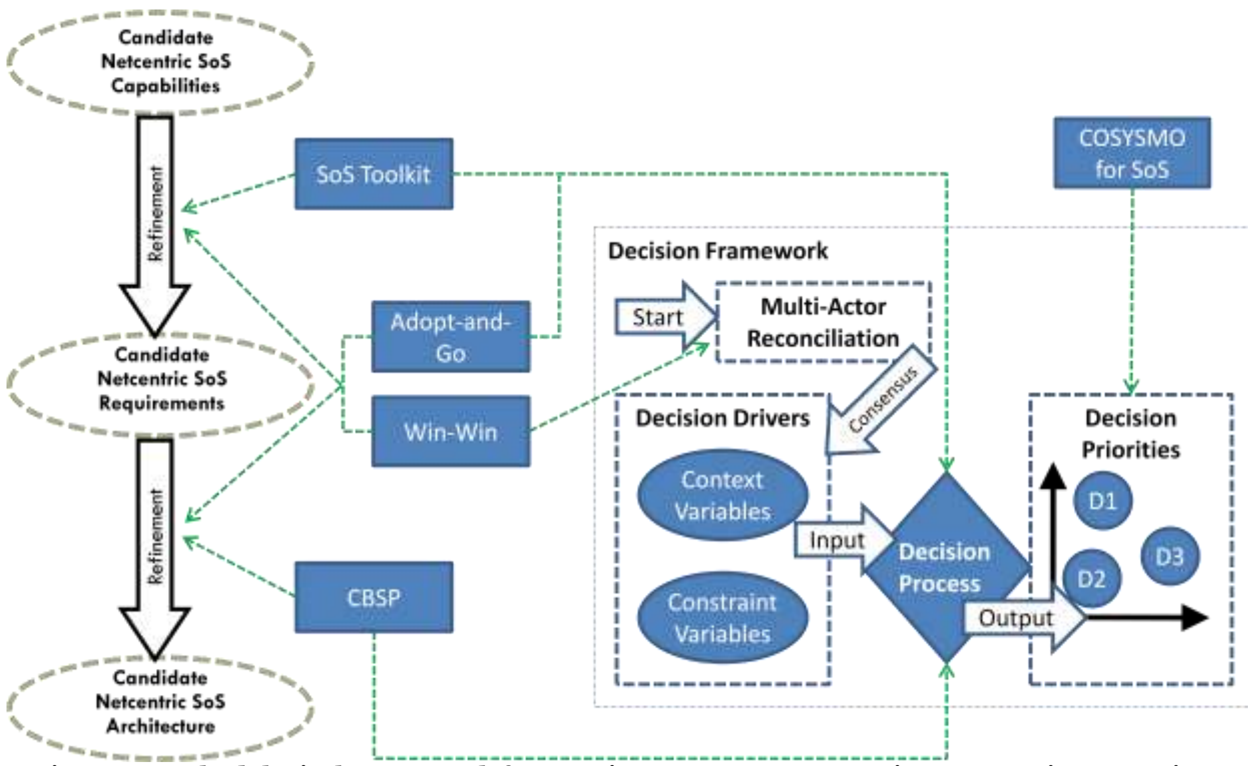

**Figure 2: Methodological Framework for Requirements Management in Net-Centric Enterprises**

The left-hand side of the figure represents the problem of managing the process of going from capability intents across the net-centric enterprise to system requirements, then from requirements to architecture designs. These are addressed further in Sections 6 and 7 of this report, respectively.

The right-hand side of the figure addresses an iterative decision process. The decisions are iterative due to the complexity of the decomposition and due to the changing mission needs over time (as well as potentially changing set of system assets and stakeholders). Thus, many decision cycles are needed in an effort to go from capabilities to architectures.

Given the multi-stakeholder nature of the net-centric enterprise, there is a multi-actor reconciliation process at the front-end. In Section 6, the multiple stakeholders negotiate during the decomposition of capabilities to requirements and assignment of responsibilities to systems. In Section 7, a structured conflict resolution process is used. Decision drivers consist of context variables (i.e., characteristics of the integration effort) and constraint variables (constraints on decision space ranging from technological to external constraints). Decision priorities include estimated benefits on the one hand versus estimated costs and risks on the other.

It should be noted that Section 8 explores the topic of multi-stakeholder decision-making in requirements management via a case study and draws conclusions about effective practices.

Within this framework, a number of MPTs have been specified and adapted for use to address specific sub-problems.

- SoS Toolkit – This consists of a number of MPTs from the system of systems domain that are adapted for use in net-centric capabilities-to-requirements engineering. These are described in Section 6.
- Adopt-and-Go – This is an approach whereby, if multiple systems or sub-systems perform the same function, a decision is made to select one of them and discard the rest, rather than integrating them. It was used in the HP-Compaq merger (Burgelman & Meza, 2004).
- CBSP – This is a methodology for deriving architecture styles from requirements via intermediate models (Grünbacher, Egyed, & Medvidovic, 2004). It originally was specified for single-system software development, but is adapted to the net-centric integration context in Section 7.
- Win-Win – This is a structured method for resolving conflicts among stakeholders regarding requirements (Boehm, Grunbacher, & Briggs, 2001). It has been adapted to utilize web technologies (Wu, Yang, & Boehm, 2009), and it is integrated into CBSP and is thus part of the MPTs described in Section 7.
- COSYSMO – This MPT addresses cost estimated for systems-of-systems and is used in Section 6.

Our approach utilizes case studies of requirements management in system integration efforts to determine the potential applicability and effectiveness of our methodology and component MPTs. Case studies are selected based on relevance to the problem context of requirements management in net-centric enterprises – autonomous multiple stakeholders with enterprise-level capability intents that must be addressed via some type of integration of systems. Example case study applications are in autonomous systems of systems, health IT, corporate mergers, private-public partnerships and regional area crisis management.

The goals of case study analysis are (i) to identify issues/challenges, (ii) to determine adaptations for MPTs, and (iii) to evaluate methodology benefits/costs. Expected outcomes are (i) a manual/tutorial for methodology, (ii) the enumeration of remaining research problems, and (iii) evidence of value to the user community.

# 5 INTEGRATION TAXONOMY

In this section, we propose a taxonomy for system integrations that can be used as a framework for understanding the characteristics of a particular effort and identifying which approaches should potentially be used to promote success. In this taxonomy, we propose three axes along which integrations and mergers can be characterized.

- Technical complexity – the extent to which technical factors complicate the integration or merging process and the associated requirements management. These include data conflicts, technology conflicts and architectural conflicts (Land & Crnkovic, 2011), as well as the number of legacies involved.
- Stakeholder alignment – the extent to which the various stakeholders are aligned with respect to the vision for and approach to the integration. Alignment is critical to successful organizational change and enterprise transformation (Rouse, 2006), which often provides the context for system integration.
- External complexity – the extent to which external factors complicate the integration or merging process or constrain the options available. Examples include market incentives, laws and regulations, treaties, taxes and tariffs, public pressure, industry standards, etc.

Note that an implied taxonomy based on technical factors has been developed and used to drive integration strategy selection (Land & Crnkovic, 2011). The strategies considered there include loose integration, merging, chose-one, and start-from-scratch. Similarly, the idea here is to develop a (more explicit) taxonomy to drive strategies and approaches, but in a broader socio-technical context.

Figure 3 illustrates the integration taxonomy in a framework similar to the enterprise transformation framework of Rouse (2006). Stakeholder alignment ranges from a monolithic stakeholder, to command-and-control, to multiple stakeholders with different interests but a common overarching goal, to stakeholders with misaligned incentives and objectives. Similarly, external complexity ranges from market incentives, to a single jurisdictional set of laws and regulations, to multiple jurisdictions, to global (which includes treaties, tariffs, etc.). Technical complexity cannot be characterized easily due to the multiple factors that comprise it. Hence, it is illustrated merely in terms of increasing complexity. More appropriately, different strategies and tools would apply for different combinations of technical issues. A better characterization is an avenue of future research.

Moving out from the center, the integration becomes more complex, and the requirements management problem becomes more difficult. Conflicting requirements become more likely as stakeholder misalignment increases. Note that as misalignment increases, the tendency for a system integration to be temporary also increases (resulting in additional requirements). Conflicts between capability intents and resulting requirements on the one hand and external constraints on the other become more likely as the external environment starts containing an increasing number of market forces and jurisdictions. Finally, increasingly complex technical issues also make requirements management more difficult.

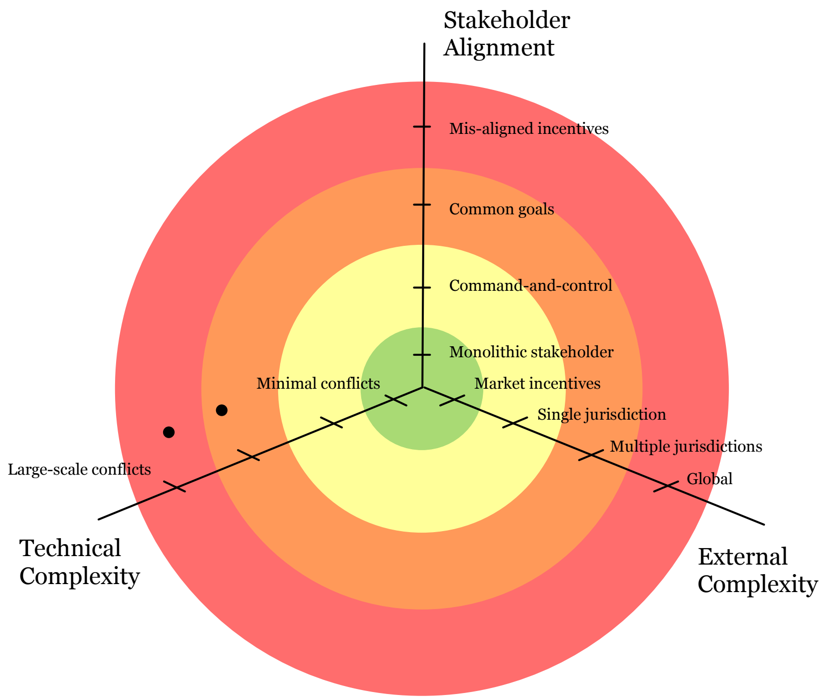


**Figure 3: Integration Taxonomy**

An initial use of this taxonomy can be done with various case studies.

- Corporate mergers – Corporate mergers result in the need to integrate or merge IT systems. The HP-Compaq merger is an example of a command-and-control situation with respect to stakeholder alignment, a complex technical challenge and a global situation with respect to external constraints (Bodner, et al., 2011). The executive team in charge of the merge was highly disciplined and firmly in control of the process. Of course, with two companies, there were many systems to be considered (especially considering that Compaq had previously merged with Digital Equipment); thus there was considerable technical complexity. The external constraints consisted of European and U.S. laws with respect to labor, monopoly practices, accounting, etc. The merger had to be approved by regulators, and this placed a number of time constraints on requirements management and system integration activities. These factors led to a successful strategy of setting up a "clean team" of personnel who were separated completely from the rest of the two merging companies (i.e., no contact). This clean team made integration decisions that were not subject to discussion or revisitation. They used two initiatives – "Adopt-and-Go" and "Launch-and-Learn" (Burgelman & Meza, 2004). In the former, if there were multiple systems that performed the same function, the clean team made a decision as to which one was kept (i.e., which system best fit the needed capabilities). Under the latter, quick action was taken that was "good enough" (i.e., no over-analysis). This led to a successful result in that critical systems were unified for the merged company on day one, while other systems were unified in the months that followed (Basole & DeMillo, 2006).

- <u>Private-public partnerships</u> – When Lockheed Martin won the contract for the F-35 Joint Strike Fighter, the contract stipulations forced a new way of doing business. In particular, Lockheed Martin had to partner with two other firms and had to operate the program on a global basis (international suppliers and customers). During the system design and development phase of the program, this meant that the partner firms had to integrate their engineering design systems to support design of the F-35. The technical issues associated with the new capabilities were highly complex, since there were numerous legacy systems, and these had to feed into a common architecture that supported a global network with almost-instantaneous data needs. There was misalignment among stakeholders. The partner firms, while united in the vision for the program, had differences when it came to approaches and also encountered reluctance in sharing proprietary methods and technologies. There was also misalignment between the private-sector and public-sector stakeholders. In terms of external complexity, the program operated on a global scale. These factors necessitated a leadership-driven process to maintain alignment across the stakeholders and support the business changes necessary to enable the technical management to develop the teaming structures for managing capabilities-to-requirements-to-systems successfully. This case study points to the importance of leadership and teaming to enable an environment that facilitates addressing technical challenges and requirement management. (This case study is detailed in Section 8.)

- <u>Regional area crisis management systems</u> – With the increasing importance of effective regional response to crises (hurricane, earthquakes, etc.), there is emphasis on integration of systems among a number of stakeholders that can help address such crises, ranging from first responders, to state and local law enforcement, to media. Lane and Bohn (2010) describe such a system-of-systems operating in Southern California, as well as the systems engineering issues and needs. While there are diverse stakeholders, they largely have a common goal of addressing a crisis. They operate under a multi-jurisdictional set of external constraints. Finally, the technical aspects of the integration are complex due to the heterogeneity of systems across stakeholders. This is an evolving system-of-systems; hence, new capabilities are often developed and deployed. Section 6 of this report illustrates that, in this type of situation, engineering tools can be applied to decompose capability intents into requirements and assignment to systems. However, there is still a substantial amount of iteration and negotiation among stakeholders involved in the process.

We argue that the outermost circle requires a transformational approach to the way business is done, similar to the framework of Rouse (2006) that focuses on scope, means and ends as axes. Today's enterprises struggle with stakeholder misalignment and global scope, plus the increasing scale and scope of technical challenges associated with modern systems. In fact, perhaps this integration taxonomy framework is an alternative way of looking at complexity and the need for transformational approaches to address complex socio-technical issues such as requirements management in net-centric enterprises.

# 6 CAPABILITIES-TO-REQUIREMENTS ENGINEERING

This section addresses capabilities-to-requirements engineering in the net-centric enterprise environment using MPTs adapted from usage in systems-of-systems research. Of course, acknowledged systems of systems have many similarities to the systems in the net-centric enterprise, and thus MPTs from that domain are generally applicable. This section considers net-centric systems as systems of systems.

## 6.1 INTRODUCTION

Given an existing set of interconnected, independent systems, often referred to as a system of systems (SoS), one of the key activities according to the DoD Systems Engineering Guide for Systems of Systems (DoD, 2008) is Translating SoS Capability Objectives into High-Level SoS Requirements. According to the DoD SoS guidebook:

> *When a formal SoS is first identified, the systems engineering team is called upon to understand and articulate the technical-level expectations for the SoS. SoS objectives are typically couched in terms of needed capabilities, and the systems engineer is responsible for working with the SoS manager and users to translate these into high-level requirements that can provide the foundation for the technical planning to evolve the capability over time. To accomplish this, the SoS SE team needs to understand the nature and the dynamics of the SoS both to appreciate the context for SoS expectations and to anticipate areas of the SoS that are most likely to vary in implementation and change over time. The SoS systems engineer has a continuous active role in this ongoing process of translating capability needs into technical requirements and identifying new needs as the situation changes and the SoS evolves.*

The DoD SoS guidebook further discusses this activity in more detail:

> *At the outset of an SoS effort, one of the first tasks facing the SoS manager and systems engineer is to develop a basic understanding of the expectations for the SoS and the core requirements for meeting these expectations. In an SoS, objectives are often stated in terms of broad capabilities. The SoS systems engineer and manager review objectives and expectations on a regular basis as the SoS evolves and changes occur in user needs, the technical and threat environments, and other areas. The SoS SE team also provides feedback to the manager and stakeholders on the viability of meeting SoS objectives, particularly given the results of other SoS SE core elements.*
> *This SoS SE core element involves codifying the SoS capability objective, which may be stated at a high level, leaving the task of clarifying and operationalizing*

*the objectives and expectations to the SoS manager, systems engineer, and stakeholders. The following examples illustrate the type of capability objectives an SoS might have:*

- *Provide satellite communications (MILSATCOM)*
- *Provide global missile defense (BMD)*
- *Provide a single view of the battle space for all customers (SIAP)*

*Once the SoS establishes the capability objective (often based upon desired operational tasks and missions), the SoS SE team defines the functions required to provide the capability and the variability in the user environment which will impact the different ways these functions will be executed. The articulation of objectives may be somewhat general at the outset, but as the SoS and SE processes mature, the objectives become more focused and they may change. 'Reference missions' or 'use cases' can be developed to evaluate the operational utility of the SoS and derive requirements that directly address usability of the SoS in the operational environment. Working with the SoS manager, users, and stakeholders, the systems engineer plays an important role in articulating capability objectives. This activity provides the systems engineer with a broader understanding of priorities and relationships, and that understanding will be useful in the further development and management of requirements. The product of this element is a set of requirements ready for incorporation to a future functional baseline for the SoS.*

*Within this core element the systems engineer develops a broad understanding of the context and drivers for the SoS. Beyond the specific functionality needs, it is very important for the systems engineer to have a good understanding of the motivation for the SoS, particularly the need to be more responsive to the increasing change tempo of the battle space, be it cyberspace, non-nation state terrorism, or health care management for veterans. Because SoS tend to evolve over time, the systems engineer needs to understand and continue to track the dynamics of change as they influence the SoS objectives and expectations. This provides the drivers for the SoS SE element 'monitoring and assessing change'; in effect, it provides the context to help the systems engineer anticipate the type of changes and variability the SoS will need to address over time.*

As illustrated in Figure 4, capability engineering starts with understanding the desired capability and identifying various options for achieving that capability. Initial capability engineering is typically done by assessing available resources and assets to identify existing functions from which the new capability can be composed (USAF-SAB, 2005), followed by a gap analysis for each alternative identified. Finally, each alternative is further evaluated in terms of capability performance, cost, and schedule resulting in information that can be used to support the trade decision.

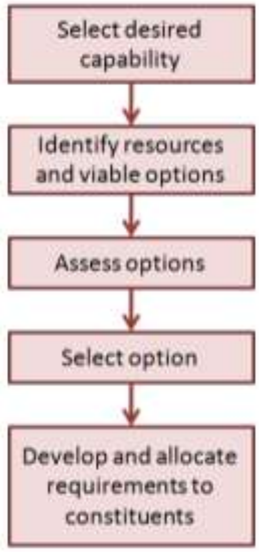


**Figure 4: Overview of Translating Capabilities into Requirements**

The goal of this research effort was to:

- Provide additional guidance for translating capability objectives into requirements,
- Define SoS engineering (SoSE) methods, processes, and tools (MPTs) that might support this activity, and
- Illustrate how the SoSE MPTs would be used and integrated to support SoS engineering using Regional Area Crisis Response SoS (RACRS) example.

While many of the techniques and methods described here are not new, they are used in ways tailored to support SoS and SoSE analyses and integrated together through a process to support capability-to-requirements engineering in a more rigorous, repeatable manner, resulting in meaningful information about alternatives that can be used to support a final decision on how the capability will be implemented. The MPTs described here are illustrated using a notional example, the Regional Area Crisis Response System (RACRS), described in (Lane & Bohn, 2010). This example has been developed to support research using actual systems in the public domain that are employed to respond to regional crisis situations (Lane & Bohn, 2010).

## 6.2 RACRS Background

The motivation for RACRS, as described in (Lane & Bohn, 2010), is based upon recent catastrophes that have happened in recent years within the United States: hurricane Katrina, devastating fires in California, powerful earthquakes in the western United States, and tornadoes in the Midwest United States. Early responders to these incidents found that communications between the different local agencies was difficult at best and often not integrated. When state and federal agencies became involved, the communications problems escalated. As a result, efforts have been initiated to establish a way to better integrate the needed agencies in response to a given incident. The goal is that the agencies will generally operate on their own outside of the SoS, then quickly be able to dynamically reconfigure and join the regional SoS as needed, typically in response to an incident.

## 6.3 Overview of Capability-to-Requirements MPTs

The Capability-to-Requirements process is illustrated in Figure 5. This figure identifies techniques and methods that can be used to support the engineering activities associated with each step.

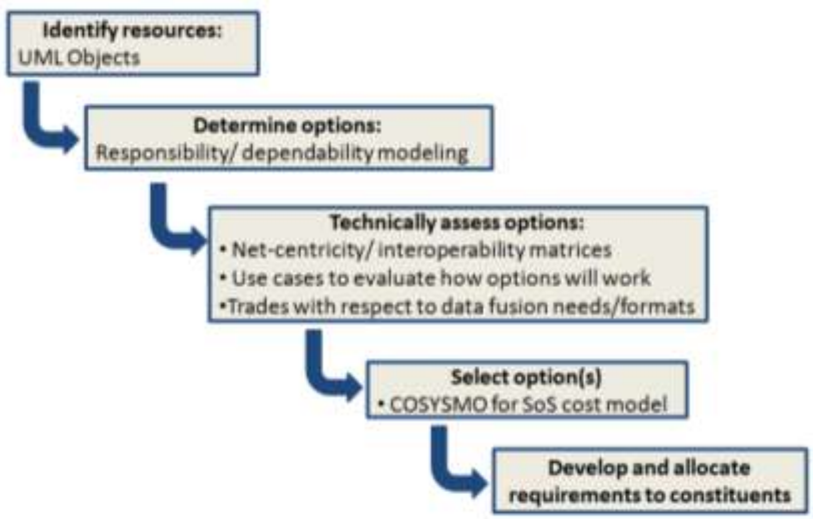


**Figure 5: Capabilities-to-Requirements Tools to Support Engineering Activities**

The following sections describe each of the tools and how they are used in the capability-to-requirements engineering process.

### 6.3.1 UML Object Models

Several UML object models are used to understand the SoS and its constituent systems as well as to identify/understand single system functions that can be used to develop new capabilities and to assess and define the various options for implementing a new capability. These models begin with “black box” models to understand the SoS and its constituents at a high level and evolve to “white box” models that capture the internal information about the constituent systems needed to understand options for various SoS capabilities. These SoS models, described in (Lane & Bohn, 2010), include:

Black box models include:

a. Context diagrams: Identify systems of interest in the SoS from selected system viewpoints as well as who and what will interact and what types of information will be passed.
b. Use case diagrams: Describe how the SoS capabilities of interest work. These diagrams can be used to model the "as is" SoS capabilities as well as alternative "to be" options for the new capability.
c. Sequence diagrams: Illustrate the sequence of requests and responses that flow within the SoS environment for capabilities of interest.

White box models include:

a. Constituent system entities: Describe the key functions and single system capabilities that can be performed by the single system.
b. Interface entities: Describe the each interface in the SoS and the information that goes over that interface.
c. Input/output (I/O) entities: Describe the details of each data element type that goes over the various interfaces.

### 6.3.2 Responsibility/Dependability Modeling

Responsibility modeling (Greenwood & Sommerville, 2011) captures information that can be used to identify socio-technical threats to the dependability of constituents in a coalition of systems or SoS. For each responsibility/capability of interest and resource/constituent system within the SoS, available resource agents that can support the capability are identified. In the second part of this modeling, the dependability of the each agent is assessed through a risk analysis process.

### 6.3.3 Interoperability Matrices

The level of interoperability between the various constituents in an SoS are captured in an $N^2$ diagram where all of the constituent systems are listed both across the top and down the left side of the matrix. Each of the other boxes in the matrix indicates the level of interoperability between each of the two systems associated with that row/column. The method used to specify the level of interoperability in the $N^2$ diagram is the Levels of Information System Interoperability (LISI) (DoD, 1998).

### 6.3.4 Data Fusion Analyses

For capabilities requiring data fusion, Solano (2011) provides guidance for trades that must be performed with respect to level of fusion, functional aggregation, producer-consumer reconciliation, incongruent or inconsistent metadata management, concept of operations with respect to the fusion(s), fusion lifecycle, network topology, and information assurance.

### 6.3.5 COSYSMO for SoS

Once viable alternatives have been identified and evaluated with respect to feasibility and risks, the final step is to evaluate the costs of implementing the various alternatives. The Constructive Systems Engineering Cost Model (COSYSMO) for SoS (Lane, 2009), illustrated in Figure 6, can be used to evaluate the relative systems engineering costs of each alternative.

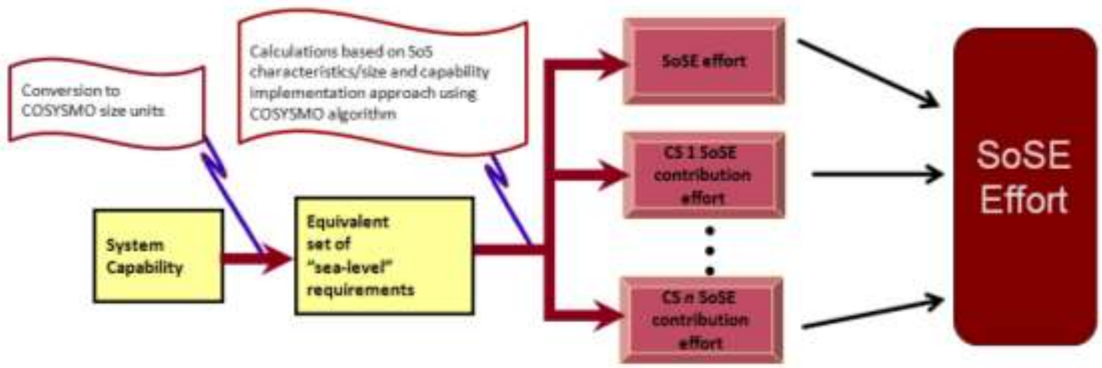


**Figure 6: COSYSMO for SoS Overview**

To develop a complete cost estimate, additional cost models in the USC CSSE Constructive Cost Model (COCOMO) family (Boehm, Valerdi, Lane, & Brown, 2005) can be used to estimate the costs of Commercial-Off-the-Shelf (COTS) integration and software development. The final cost estimate must also include the costs of hardware procurement/manufacturing that might be required for the alternative of interest.

## 6.4 RACRS Analysis using Capability-to-Requirements MPTs

For the purposes of this example, the current desire is to enhance the RACRS to provide the following improved/new capabilities:

1. Improve number of fire-fighting resources available to fight major fires in the region. (Currently RACRS is limited to local civil responders augmented with

available civil responders from other areas as well as low-risk inmates from local prisons/jails.)
2. Further reduce the time and number of official crisis management personnel resources required to evacuate a specified area/region. This capability includes the ability to quickly determine areas that require evacuation and appropriate evacuation routes as well as the ability to evacuate large numbers of people that do not have transportation (e.g., assisted living residences, hospitals, jails).
3. Protect evacuated areas from looters.

At the same time, the RACRS stakeholders want to:

1. Minimize local government expense (city, county).
2. Minimize risk to human life (crisis responders and local population).
3. Minimize workload on skilled personnel responsible for responding to crisis.

The following identifies potential resources that might be used to provide improved/new capabilities:

1. Local: fire fighters, police, and sheriff personnel.
2. Volunteer civilians.
3. Military personnel at local bases.
4. Low-risk inmates incarcerated in local jails.
5. Unmanned aerial vehicles (which require people to remotely operate).
6. Unmanned ground vehicles (which require people to remotely operate).
7. TV/radio station announcers.
8. Satellite and local road camera images showing crisis area (e.g., fire) and traffic status.
9. Buses for transporting people.
10. New system: Reverse-911 system that calls homes/residents of given area and tells them when, how, and where to evacuate to via pre-recorded messages.
11. Homeowner alarm/security systems to notify people of need to evacuate, notify security patrols of break-ins, potential looters.

The following illustrates how the above MPTs might be employed to develop a set of requirements to fulfill the desired capability improvements.

### 6.4.1 Identify Resources

The constituents of interest for this version of the SoS are those described in (Lane & Bohn, 2010):

- <u>Satellite imaging system</u>: Provides images of interest to requestor.
- <u>Fire department</u>: Manages the fire response units.

- Police department/sheriff's department: Provides safety and crime-fighting support that includes evacuation support and protection from looters.
- Handheld devices: Provides connectivity to crisis responders on the ground via voice and video.
- Reverse-911: Automatically sends voice messages to people that reside or work in areas that need to be immediately evacuated.
- Regional area planning and land use data: Includes building plans, building locations, and maps for utilities (electricity, water, sewer) for regional areas of interest.
- Unmanned aerial vehicles (UAVs): Used for surveillance, lightweight fire retardant drops, and can also be armed to start needed backfires or fire upon looters/rioters.
- Unmanned ground vehicle (UGV): Provides
    - on the ground video feeds in situations where it is too dangerous for personnel,
    - clearing of brush/small trees to create fire breaks.
- Aerial water tanker: Canadian asset shared among multiple U.S. regional areas to drop water on hot spots.
- News helicopter: Used to capture video feeds for news programs—includes news events as well as traffic flows and may also be used to monitor for signs of looting.
- Command and control center (CCC): Central site to monitor and help coordinate activities support decision makers.

The constituent systems that interoperate with the CCC for the RACRS fire-fighting scenario are illustrated in the CCC context diagram shown in Figure 7 (Lane & Bohn, 2010). This is the "black box" view of the CCC and is used to understand at the top level the constituent systems related to fire-fighting from the CCC point of view.

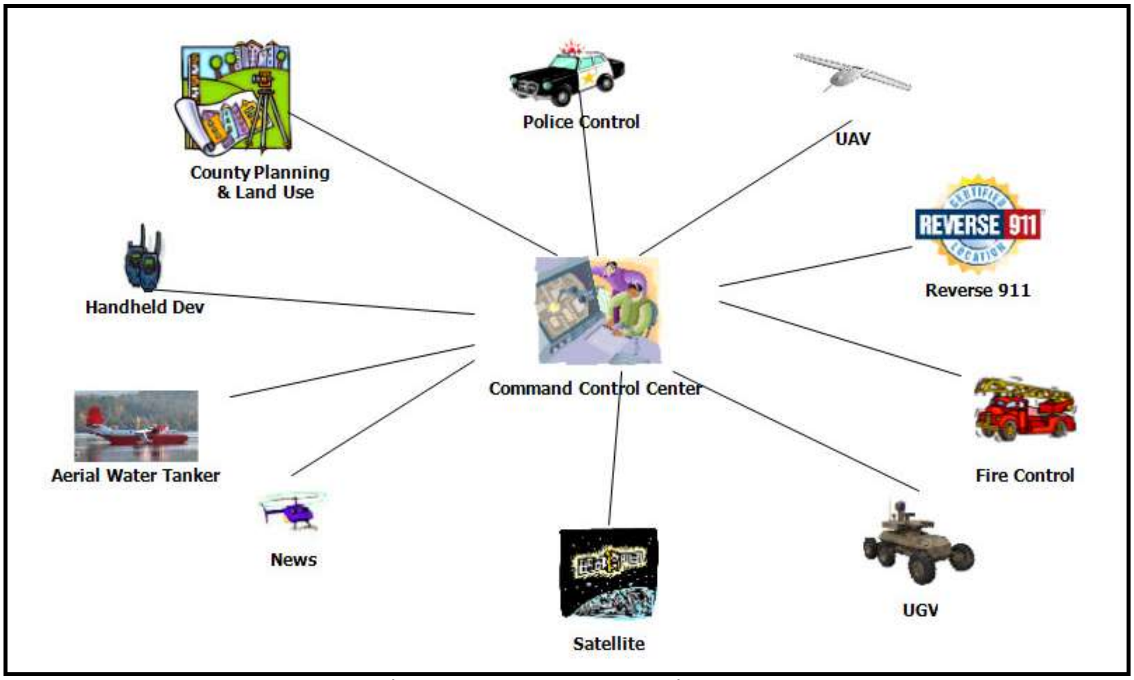


**Figure 7: CCC Context Diagram**

### 6.4.2 Identification of Capability Alternatives

The initial analysis evaluates the feasibility of utilizing military resources in the region to support firefighting and then focuses on improving the manpower requirements needed to support the evacuation process. Potential alternatives to consider for the improved evacuation capability are:

1. Use current local patrols (police/sheriffs) (e.g., personnel employing loudspeakers, roving patrols, roadblocks).
2. Use civilian (volunteer) patrols (e.g., personnel employing loudspeakers, roving patrols, roadblocks).
3. Use unmanned vehicles (combination of ground and aerial that can warn of potential harm/record suspicious activities.
4. Rely on homeowner alarms to notify patrols.
5. New reverse-911 system that can be used to automatically notify residents in a given area to evacuate.

### 6.4.3 Analysis of Alternatives

The various MPTs are illustrated below and show how they can be used to support the analysis of capability alternatives.

**Responsibility/Dependability Analysis**: To start the analysis of alternatives, a responsibility matrix is developed that lists the various capabilities in the left-hand column and the potential resources across the top, as illustrated in Table 1.

| Responsibility | Resources | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Fire trucks | Sheriff cars | Water tankers | UAV | Reverse 911 system | Ambulances | Buses | Manual |
| Fight fire | Local<br>Regional<br>Military | Local | Canadian company | Military | | | | Local<br>Regional<br>Military<br>Volunteer<br>Low-risk inmates |
| Evacuate homes and businesses | Local<br>Volunteer | | | | Local CCC personnel | | | |
| Evacuate assisted living homes | Local | Local | | | Local CCC personnel | | Local transit<br>Charter | Assisted living staff<br>Volunteers |
| Evacuate hospitals | | | | | | Public<br>Private | | Hospital staff<br>Volunteers |
| Prevent looters | | Local | | Military | | | | |

**Table 1: RACRS Evaluate Area Responsibility Matrix**

Next, the various resources are evaluated with respect to their dependability to support each responsibility. For those resources that may not be fully depended upon, risks are defined and documented in a dependability risk table, illustrated in Table 2. The following describes the various columns in the table based on (Greenwood & Sommerville, 2011):

- Risk: an identifier
- Target: the specific resource
- Hazard: a selection from a defined set of keywords – a candidate list in (Greenwood & Sommerville, 2011) is:
    - Early – performs before required
    - Late – performs after required
    - Never/unavailable – never performs
    - Incapable – attempts to perform, but not capable of completing
    - Insufficient – performs, but at an insufficient level
    - Impaired – performs incorrectly
    - Changes – responsibilities permanently change
- Condition: describes the condition that might occur as a result of the hazard

- <u>Consequences</u>: impact of condition resulting from hazard
- <u>Severity</u>: level of impact resulting from hazard

Table 2 identifies some candidate risks associated with the resources identified in Table 1 above. Note that this table is not comprehensive, but used to illustrate the MPT.

| Risk | Target | Hazard | Condition | Consequence | Severity |
|---|---|---|---|---|---|
| 1 | Regional fire trucks/fighters | Late or never due to fires in own region | Reduced fire-fighting capability | More extensive fire damage | Medium to high, depending on other available resources |
| 2 | Canadian water tanker | Late or never due to other commitments | Reduced fire-fighting capability | More extensive fire damage, longer to put fires out | Medium to high, depending on other available resources |
| 3 | Local fire trucks | Unavailable due to reallocation to other fire | Reduced fire-fighting capability | More extensive fire damage, longer to put fires out | Medium to high, depending on other available resources |
| 4 | Reverse 911 System | Insufficient | Not all residents notified to evacuate | Residents at risk for being trapped/ affected by crisis (fire, hazardous material, etc.) | Low to high, depending on type of crisis requiring evacuation |
| 5 | Low-risk inmates | Various – may be unskilled, may escape custody | Fire-fighting capability is less than that of experts | Additional resources required to train and monitor inmates | Low severity with respect to crisis, but medium severity with respect to costs associated with training and monitoring |

**Table 2: Dependability Risk Matrix**

**Interoperability Assessment**: This MPT assesses the ability of the relatively dependable systems to interoperate with each other. The first step to understanding and managing interoperability within an SoS is understanding the information that flows across each interface and the format of the data elements that are part of that information. Figure 8 (Lane & Bohn, 2010) illustrates how UML interface and input/output (I/O) entities can be used to capture this information for key RACRS interfaces.

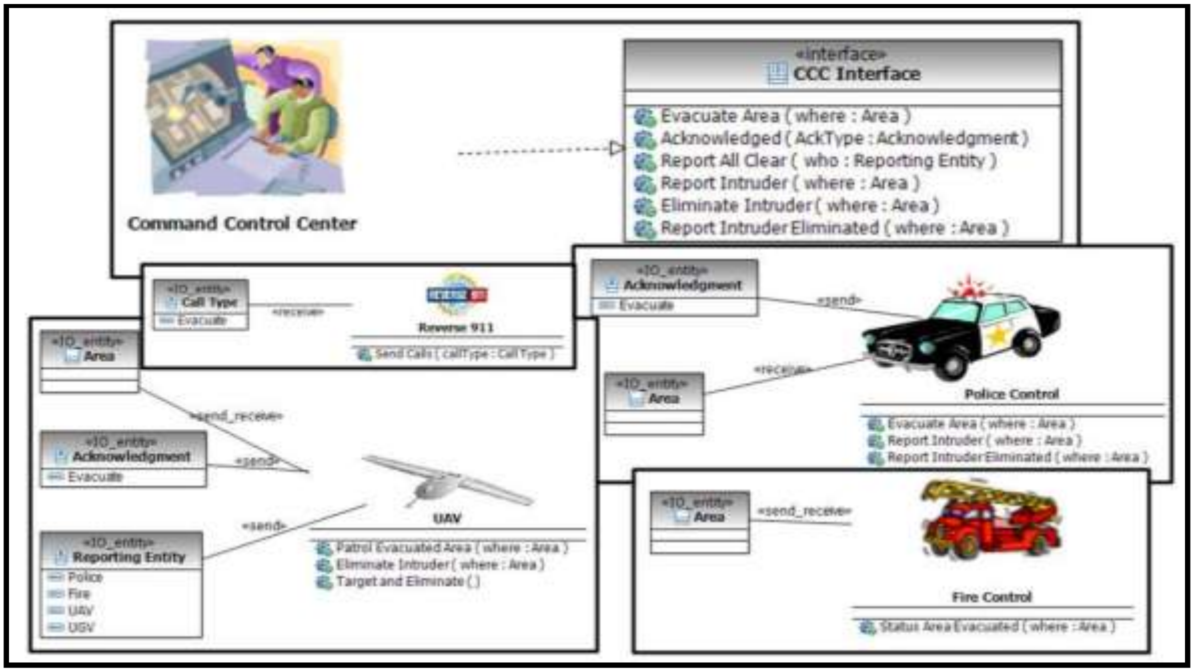


**Figure 8: Interface Class and Evacuate Area I/O Entities by Actor**

The next step is assessing the interoperability of the various constituent systems. For this assessment, an N² matrix is developed where the systems are listed both down the left column and across the top. Then each pair of systems is evaluated for interoperability using the LISI model (DoD, 1998). The level of interoperability can be specified for each of four PAID attributes: Procedures, Applications, Infrastructure, and Data. The levels of interoperability that can be specified for each attribute using in the LISI model are isolated, connected, functional, domain, and enterprise (DoD, 1998). Table 3 shows a Data interoperability matrix for the RACRS firefighting systems.

| Fire-fighting Constituents | Local | Regional | Military | Canadian | Volunteer | Low-risk Inmates |
|---|---|---|---|---|---|---|
| Local | | | | | | |
| Regional | Functional | | | | | |
| Military | Isolated | | | | | |
| Canadian | Connected | Connected | Isolated | | | |
| Volunteer | Connected via handheld devices | Isolated | Isolated | Isolated | | |
| Low-risk Inmates | Connected via handheld devices | Isolated | Isolated | Isolated | Connected via handheld devices | |

**Table 3: Firefighting Data Interoperability Matrix**

Note that the cells on the diagonal are shaded. This reflects the fact that every system should be fully interoperable with itself (if not, then these cells should contain an assessment value). Also note that in many cases, system interoperability is bi-directional and in these cases, one only need assess the systems interoperability above or below the diagonal (but not both). If interoperability is not bi-directional, then the full matrix should be completed.

**Use Cases**: For those systems that are evaluated as reasonably dependable and interoperable, use cases are developed to show how the systems will interact to perform the various desired missions. Figure 9 illustrates a top level use case diagram for the key RACRS mission/support missions (evacuate area, conduct fire suppression, get real-time topographical map info, and get static map info).

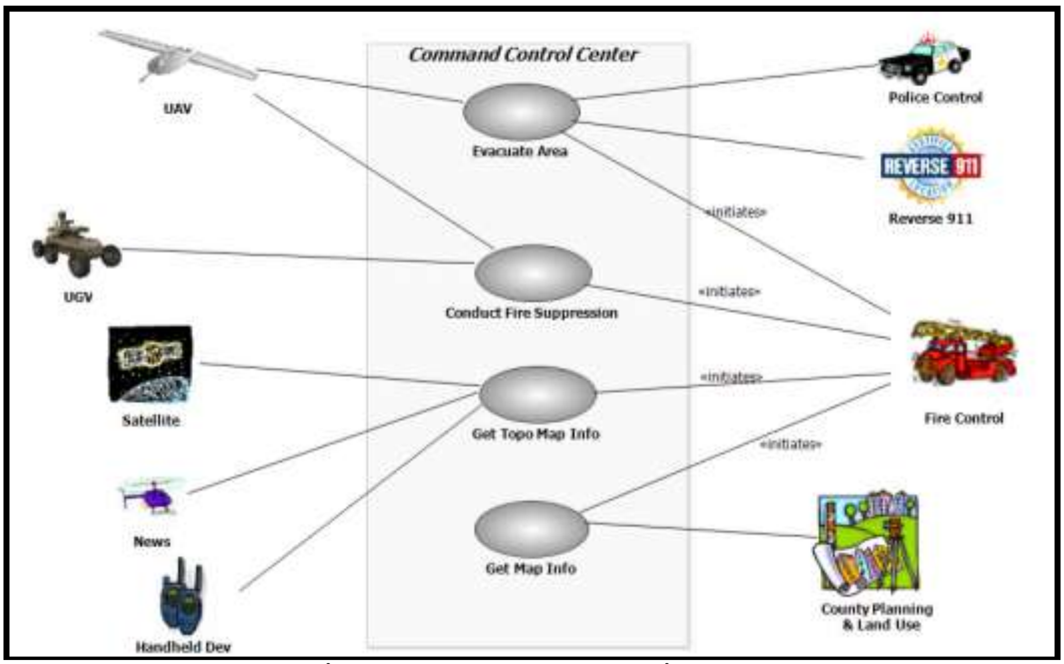


**Figure 9: RACRS Use Case Diagram**

Each use case can be further refined and analyzed by developing sequence diagrams, as illustrated in Figure 10 for the Evacuate Area mission. This diagram shows the interactions between the various constituent systems in performing the Evacuate Area mission/support mission.

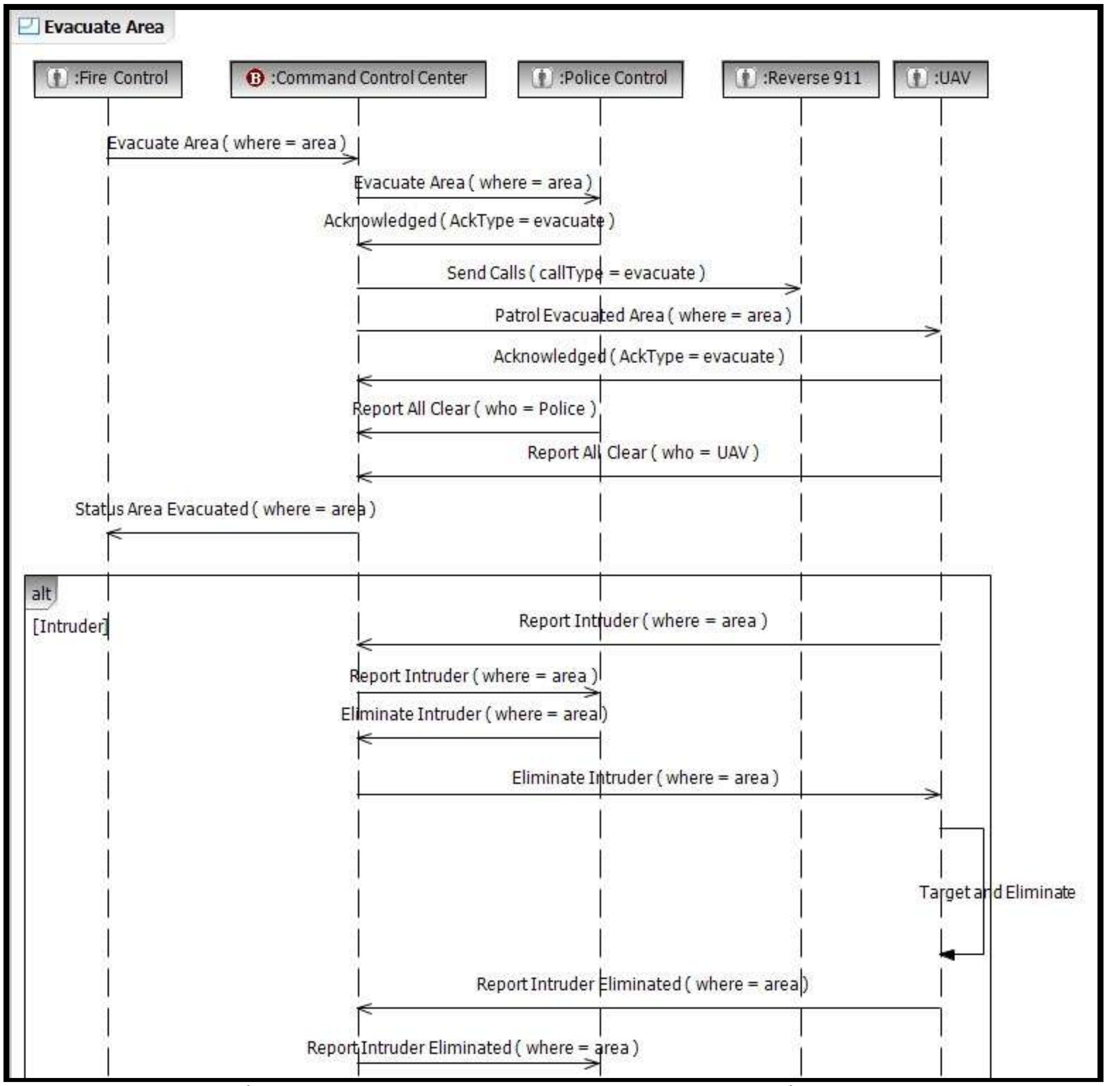


**Figure 10: RACRS Evacuate Area Sequence Diagram**

## 6.4.4 Identifying and Implementing SoS Requirements for Selected Alternative

Once the various capability options are assessed, a solution is selected and then an appropriate set of requirements is developed for each of the constituent systems that need to be modified to provide the desired capability. So, in the case of the Evacuate Area capability, an assessment might determine that it would significantly improve the capability to acquire the Reverse 911 system and to integrate it into the RACRS CCC. The final step in this process would be to evaluate the cost of the acquisition of the Reverse 911 system and the integration of that into the CCC to determine if there would be a reasonable return on investment. The costs would primarily consist of the cost of the system, the systems engineering effort to develop and test an integration approach to the CCC, then any software development and test costs needed to implement the interface.

## 6.5 Conclusions and Next Steps

The research described in this section shows that existing MPTs can easily be re-purposed and used together to support SoS capability to requirements engineering, resulting in a fairly rigorous technical analysis of capability options and the costs required to implement each . The next steps are to continue to refine these MPTs through the analysis of more complex capabilities and to further investigate and refine the data fusion MPT. This work is left for future research.

# 7 REQUIREMENTS-TO-ARCHITECTURES ENGINEERING

## 7.1 INTRODUCTION

Once a reasonably complete set of system capabilities and requirements are agreed upon by the (system-of-)system stakeholders, the capabilities and requirements will begin to drive the architecture. This process will occur iteratively, meaning that, in turn, the architectural decisions and choices made will result in the elicitation of new requirements. The two outer "peaks" in Figure 11 are part of the Twin Peaks model suggested in literature for relating requirements and architecture (the middle peak is not part of the originally proposed model and will be discussed further below). The Twin Peaks model suggests that requirements and architectures are evolved iteratively and concurrently.

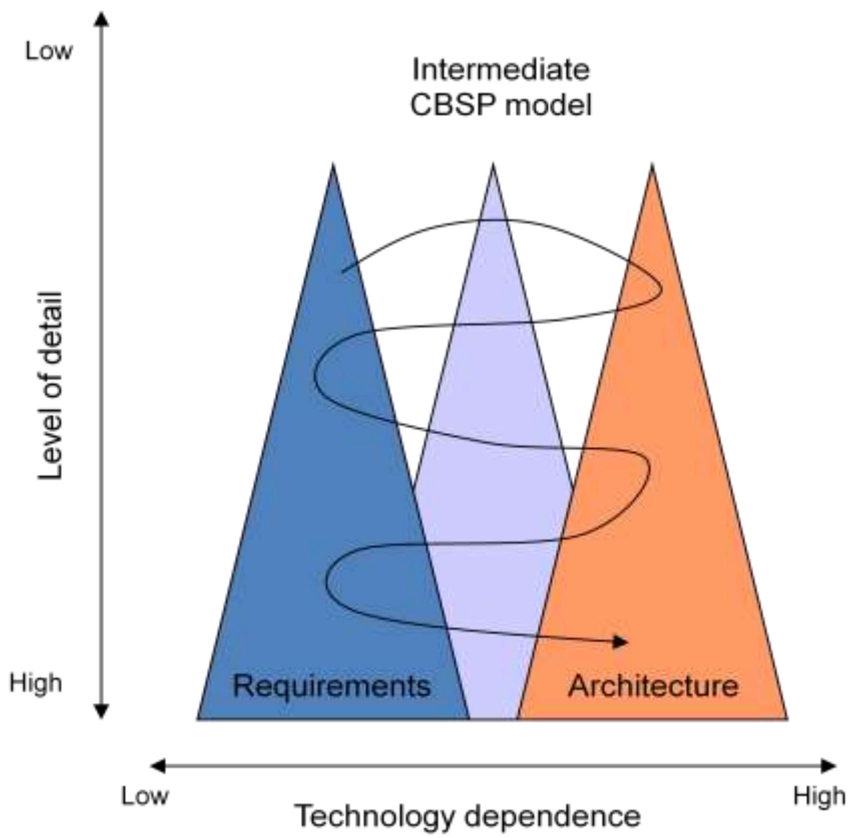


**Figure 11: The CBSP in the Context of the "Twin Peaks" Software Development Process**

## 7.2 ICBSP

The CBSP (Component-Bus-System-Property) approach helps to refine a set of requirements by applying a taxonomy of architectural dimensions (Grünbacher, et al., 2004). The intent is to provide a generic approach that primarily works with arbitrary informal or semi-formal requirements representations as well as different architecture modeling approaches. Although requirements may also be captured in a formal language (e.g., KAOS), informal or semi-formal approaches are still used very frequently. In particular, CBSP has been integrated with the WinWin requirements negotiation approach, which supports multi-stakeholder elicitation of requirements and captures requirements informally but in a structured fashion.

CBSP provides an intermediate model between the requirements and the architecture that helps to iteratively evolve the two models. For example, a set of incomplete and quite general requirements captured as statements in a natural language might be available. The intermediate CBSP model then captures architectural decisions as an incomplete "proto-architecture" that prescribes further architectural development. The intermediate model still "looks" like requirements but "sounds" like an architecture. The CBSP approach also guides the selection of a suitable architectural style (e.g., client-server, peer-to-peer, layered, dataflow, etc.) to be used as a basis for converting the proto-architectures into an actual implementation of a software system architecture. In such a context, the intermediate CBSP model can be used at different levels of detail in the modeling process. For example, it can help to refine high-level, informal requirements early in a project and more elaborated requirements in later iterations; or it can also help to understand how issues arising in architecture modeling and simulation relate to the requirements. The middle "peak" in Figure 11 depicts this intermediate nature of CBSP.

CBSP provides:

- a lightweight way of refining requirements using a small, extensible set of key architectural concepts;
- mechanisms for "pruning" the number of relevant requirements, rendering the technique scalable by focusing on the architecturally most relevant set of artifacts;
- involvement of key system stakeholders, allowing nontechnical personnel (e.g., customers, managers, even users) to see the impact of requirements on architectural decisions if desired;
- and adjustable voting mechanisms to resolve conflicts and different perceptions among architects.

Together, these benefits afford a high degree of control over refining large-scale system requirements into architectures, via a five-step process:

1. Selection of requirements for next iteration — based on importance to project success and feasibility with respect to technical, economic, organizational, or political constraints on implementing a requirement.
2. Architectural classification of requirements — each requirement is rated by the stakeholders for its relevance to the system's **C**omponents, **B**uses (i.e., connectors enabling component interaction), the entire **S**ystem, or system **P**roperties.
3. Identification and resolution of classification mismatches — any inconsistencies in how the stakeholders perceive individual requirements' relevance to system architecture must be discussed and resolved.
4. Architectural refinement of requirements — each requirement is restated into multiple C, B, S, and/or P statements, based on the requirement's relevance to those dimensions.
5. Trade-off choices of architectural elements and styles with CBSP — multiple styles may be possible for a given problem, and the system architects must select the one that will maximize the system's utility to the stakeholders, while minimizing any issues introduced by the selected solution.

While useful in "greenfield" development contexts, by itself CBSP does not provide explicit support for the context of integration, mergers, and interoperability. In that context, the four ingredients of CBSP gain a different meaning:

- C now becomes the set of constituent systems to be merged/integrated;
- B becomes the set of available integration and interoperability mechanisms;
- S is the merged/integrated system or SoS; and finally
- P describes the relevant properties of the above.

In addition, some of the requirements related to integration define the technical context (e.g., the interface of an existing system complies with the REST architectural style (Fielding & Taylor, 2002)) and/or constraints (e.g., IBM's middleware technologies must be used for communication) of the integration. Hence, iCBSP is expanded with another category that captures requirements that define the technical the technical context and constraints.

Our objective in this project has been to refine integration requirements into an integration architecture (or proto-architecture), while retaining strong traceability between the proto-architecture and requirements. On the surface, the idea behind iCBSP remains the same as that of CBSP. However, the details and execution of the two differ substantially. Specifically, iCBSP entails the following steps:

- Pre-Step: The initial activity in the iCBSP process is for the stakeholders to filter out the set of requirements that are explicitly related and relevant to integration. The other requirements are unimportant for this purpose.

- Step 1: The system stakeholders rate the importance and feasibility of the selected integration requirements.
- Step 2: The architects then rate architectural relevance of these requirements. Note that important requirements (selected in the previous step) may be deemed architecturally irrelevant, while the less important ones may have significant architectural relevance.
- Step 3: The process in step 2 will occasionally result in different views of a given requirement's architectural relevance. The reasons include the architects' differing perceptions of the system integration needs, but also the differing understandings of what the requirements themselves entail. For this reason, the architects will need to negotiate and reconcile their disagreements.
- Step 4: Finally, the final set of the architecturally relevant requirements will be broken down into C-, B-, S-, and P-relevant constituents, rephrased as needed, and then mapped to the proto-architecture.

While this set of steps appears to be a simple linear process, in practice there is significant iteration between steps due to negotiation among stakeholders.

The equivalent of this last step (Step 4) is pursued in the original CBSP approach by using "hints" that are present in the capabilities and requirements, the characteristics of the problem (application) domain, the architects' experience, and the architectural lessons-learned from and characteristics of previously implemented similar systems (e.g., selected architectural styles or patterns, middleware platforms for system implementation, etc.). However, these hints are not going to be particularly useful in the case of iCBSP. For example, the architectural styles that CBSP relies on—such as dataflow, client-server, publish-subscribe, peer-to-peer, and so on—are primarily intended for greenfield development scenarios. Therefore, iCBSP required a new set of architectural guidelines and "hints", which led to our work on the system integration matrix, described next.

## 7.3 Integration Matrix

The integration architecture obtained using the iCBSP process defines only the elements of the desired system-of-systems, while deferring decisions on the specific technical solutions that implement the integration. To facilitate knowledge capture and decision making related to the technical integration solutions, we have devised a novel concept called an integration matrix. An integration matrix is a knowledge capture method that allows representation of the effect (compatibilities, conflicts, or other relationships) certain technical solutions have on the properties of an integrated system. The column headers of an integration matrix should be labeled with the specific technical solutions – i.e., alternative or recurring design options (e.g., patterns, styles, data management solutions, COTS product combinations). The rows of an integration matrix are labeled with the potentially desired properties and outcomes of interest in an integrated system (e.g., quality goals and non-functional properties, commonly required features). The cells of an integration matrix capture the relationship between the alternative design options and the system properties; the relationship can be represented with a concise symbol (e.g., +/- to capture desirable or undesirable relationships) or rank (e.g., 1 to 10). An important feature of our work on integration matrices is that we propose further capturing these relationships with links to textual documentation pertaining to the relationship (e.g., expert rationale of past experiences).

In our work, we were specifically focused on capturing knowledge on how a general integration style of system integration affects the integrated system's non-functional properties. An integration style is a set of rules that guides the composition of the systems being integrated into an integration architecture. Note that multiple integration styles can be used in a SoS. To capture the relation of an integration style element and the candidate non-functional properties of an integrated system, we have developed an integration styles matrix depicted in Table 4.

The columns of the integration styles matrix are labeled with the candidate options for the three dimensions that comprise an integration style: connector roles, topology, and linkage mechanisms. The connector role dimension refers to the features that a connector that integrates two systems needs to provide (e.g., protocol adaptation is required to enable communication between heterogeneous system interfaces). Topology defines the geometrical structure of the integration architecture (e.g., hub and spoke topology has a central hub that controls and routes the communication between all integrated systems). The linkage style dimension defines the means through which the integrated systems communicate (e.g., the systems may communicate through explicit references that they have of each other). Note that the common integration styles (e.g., service-oriented-architecture, federated database) can be defined using the different options for style dimensions. For example, in its most basic form SOA uses distributor connectors combined with shared bus topology, while the integrated systems communicate via asynchronous messages. Similarly, federated databases use arbitrator connectors for database access, and the central database is the hub of hub and spoke topology, while the concept of a federated database equates to a shared data repository.

The cell values of the integration styles matrix define how selecting a particular element of an integration style would affect a non-functional property of interactions between two integrated systems and the system-of-systems as a whole. The cell values stand for positive effect (+), negative effect (-), neutral effect (0), and positive or negative effect, depending on the specific of the integration problem at hand (+/-). These high-level relationships should, in general, be accompanied with expert knowledge, past experience, and rationale that explain the relationship in natural language. We have collected and documented the rationale for the relationships represented in the integration styles matrix.

| **Integration styles vs. Properties** | | **Topology** | | | | **Linkage** | | | | **Connector** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Point-to-Point | Hub and Spoke | Shared Bus | Peer-to-Peer | Shared Data | Messg. | Explicit invoc. | Data Strm. | Adapt. | Transl. | Arbitr. | Distr. |
| **Interaction** | Distributed | o | + | + | + | o | + | + | + | o | o | + | + |
| | Local | o | - | + | - | + | o | + | + | o | o | o | - |
| | Secure | + | - | o | +/- | - | o | o | o | o | o | + | - |
| | Data intensive | + | - | - | + | + | - | o | + | o | - | + | + |
| | Data formats incompatible | o | + | o | - | - | + | o | o | o | + | o | o |
| | Data consistency | o | + | o | - | + | o | o | - | o | o | + | o |
| | Interaction protocols incompatible | o | + | o | - | + | o | - | o | + | o | o | o |
| | Reliable | + | - | + | + | - | + | + | o | o | o | + | o |
| | Real time | + | - | +/- | - | + | - | + | + | o | o | + | o |
| | One-to-many | - | + | + | + | +/- | + | - | + | o | o | + | + |
| | Many-to-one | - | + | o | +/- | o | + | - | o | o | o | + | + |
| | Always available | + | - | o | + | - | + | o | o | o | o | + | o |
| | Periodically scheduled | + | o | o | - | o | o | o | o | o | o | + | o |
| **System** | Loose coupling | - | + | + | +/- | - | + | - | - | + | + | + | + |
| | Robustness | - | - | + | + | - | + | +/- | - | o | o | + | + |
| | Dynamically reconfigurable | - | o | + | + | o | + | + | o | + | + | + | o |
| | Scalable | - | - | + | + | - | + | o | o | o | o | + | + |
| | Caching | - | + | + | o | + | o | - | - | o | - | + | + |
| | Distributed transactions | - | + | + | +/- | + | + | + | o | o | o | + | + |

**Table 4: Integration Styles Matrix**

To facilitate adoption of integration matrices, we propose utilizing the modern Wiki-technologies that allow the engineers (1) to quickly create their own matrices, (2) to link the relationship captured in the matrix cells to additional pages containing rationale, and (3) to input their own rationale based on past experience. The Wiki implementation of the integration styles matrix is depicted in Figure 12. Note that the cells in the matrix are highlighted in green because they link to rationale pages. For example, following the link for the negative relationship between Hub and Spoke and Data intensive property would open an additional page with rationale suggesting that "Hub quickly becomes the bottleneck of the system integration." In our case studies, we have evaluated how the integration matrix can be used not only to capture such knowledge, but to quickly detect potential integration problems, make informed decisions based on this knowledge, and arrive at a superior integration solution.

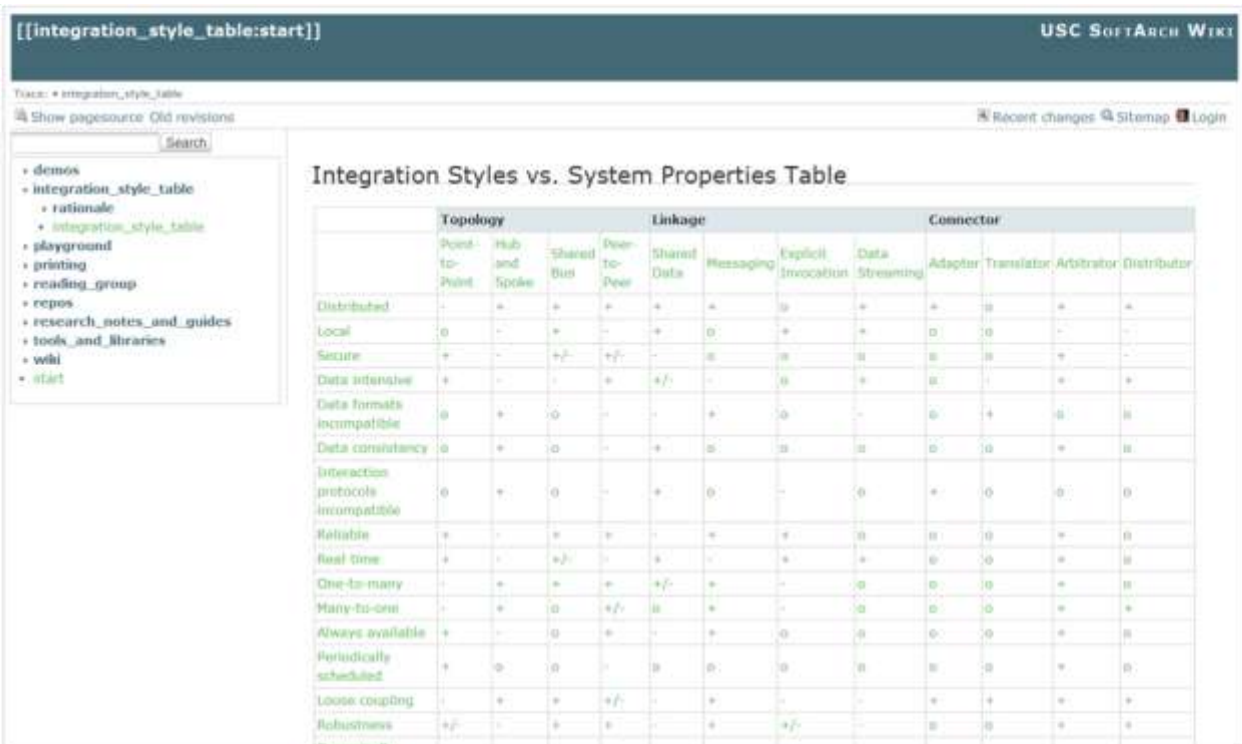


**Figure 12: Integration Styles Matrix Wiki Implementation**

## 7.4 Case Study

To validate our research on refining integration requirements into an integration architecture, we applied iCBSP and the integration style matrix on Jail Information Management System (JIMS). The case study involved (1) applying the iCSBP steps on the available JIMS requirements, and (2) retroactively applying the integration styles matrix on JIMS, while analyzing how the matrix could have prevented the documented issues that occurred during the implementation of JIMS.

JIMS is a system that provides data consistency across seven San Diego County detention centers. JIMS is also a part of the Regional Area Crisis Response System-of-systems (in Section 6) and had to be integrated with a number of external systems, including the Field Reporting System, Warrant Search Systems, Criminal History System, Court Management System, and Computer-Aided Dispatch System. The reasons that JIMS made for a particularly relevant and insightful case study are fivefold: (1) the availability of the full JIMS requirements (1800 original requirements), (2) JIMS is integrated at multiple levels (integration with external systems, integration across the seven detention centers, as well as integration of COTS in the individual JIMS sites), (3) the availability of parts of the JIMS architectural documentation that describe the issues with the JIMS implementation, (4) the numerous required non-functional qualities (e.g., security, privacy, performance, reliability, and availability), and (5) access to a technical expert who worked on the design of JIMS.

## 7.5 iCBSP Applied to JIMS

To apply iCBSP to JIMS, we first filtered the overall set of JIMS requirements for integration-relevant requirements. This pre-step resulted in 46 requirements, which we used for the subsequent iCBSP application. In the first iCBSP step, we had two stakeholders rate the importance and feasibility (0 to 3) of each of these requirements. In case the disagreement between the stakeholders was high, the stakeholders had to engage in WinWin negotiation to decide on the appropriate ratings (as a disagreement, we considered a cumulative difference of more than 3 points in importance and feasibility). For example, one requirement was, "*To minimize the amount of retraining required by booking clerks and other personnel, the booking screens used in JIMS shall be similar, to the extent feasible, to IBIS screens*." The stakeholder disagreement for this requirement was high because of divergent importance ratings: a technical stakeholder considered the requirement to be of low importance as she interpreted it as a UI issue, while the end user rated this requirement highly as it described an important aspect of interoperability with a legacy system. Eventually, both stakeholders agreed on the importance of this requirement in the context of system integration. Applying the first step of iCBSP has proven to effectively filter out the irrelevant and unfeasible requirements from 46 to 30 requirements. Furthermore, WinWin negotiation was applied in 11 cases that were initially inconclusive.

In the second iCBSP step, we had two system architects rate the architectural relevance of 30 important and feasible requirements; this is done in terms of each requirements relation to CBSP dimensions. The rating process resulted in strong agreement among the architects on nine requirements, two of which were deemed irrelevant from an architectural perspective. For example, one requirement was, "*Vendor shall assist the*

*Sheriff's Department in the design, planning and implementation of the data conversion from the existing IBIS system. A conversion team will be formed consisting of vendor and Sheriff staff, and vendor shall oversee the implementation and assist Department staff in problem diagnosis and conversion issues.*" This turned out to be a staffing issue rather than an architectural issue.

For those requirements that were not agreed upon, the architects had to reconcile their views. For example, the architects had to negotiate on one requirement involving the Records and Information Management System (RIMS) – "*When the RIMS criminal history module initially becomes available, JIMS shall interface with both it and the Records Index and the Criminal History Systems. This dual interface shall be maintained until such time as all records in the Records Index and the Criminal History Systems have been moved into RIMS, or the records remaining in the Records Index and the Criminal History Systems are judged to be sufficiently old as not to be of value except in special cases.*" The issue was whether this relates to a property of the individual integrated systems. The requirement suggests that the integrated systems need to maintain the records; however, since this is required from multiple integrated systems and their interconnections, the requirement was eventually marked as a system property (data availability) requirement. The third iCBSP step helped to additionally filter out architecturally irrelevant requirements (8 requirements filtered out), while also relating the requirements to specific integrated architecture elements.

As a part of the fourth iCBSP step, we graphed the mapping between the integration requirements and the elements of the integrated system-of-systems architecture. Figure 13 depicts an example graphing for one of the JIMS requirements; note that this example illustrates the important iCBSP features: explicit relation and traceability between the requirements and the elements of the integrated architecture. Once the mapping is completed, iCBSP mandates rewriting overly complex requirements.

The final result of applying iCBSP to the JIMS case study was a set of 16 identified systems that need to be integrated, 16 interconnections between JIMS and those systems, and 11 data component elements. The final traceability graph had over 100 traces. Overall, iCBSP has proven helpful in effectively filtering the requirements, creating traceability links to an integration architecture, and improving the quality of the requirements themselves.

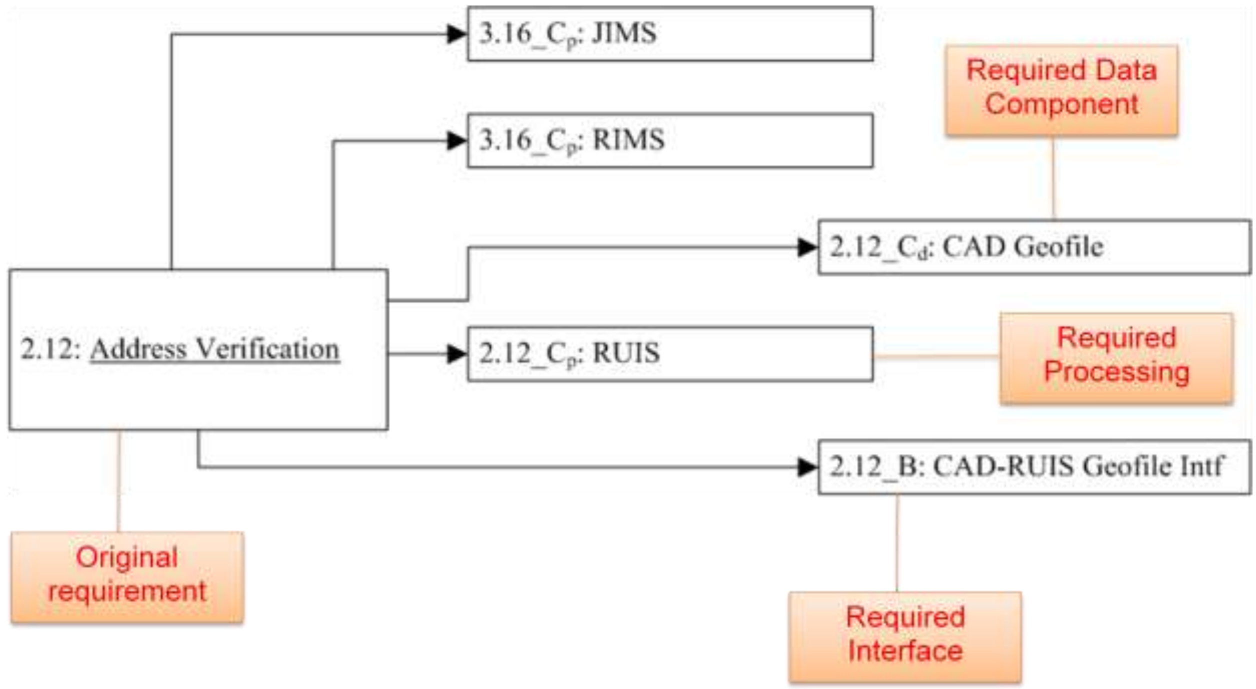


**Figure 13: The mapping between one of the requirements and the integrated architecture elements**

## 7.6 Integration Styles Matrix Applied to JIMS

To assess the benefits of the integration styles matrix that we developed, we have applied it at multiple levels of JIMS. In this section, we illustrate the application to the JIMS cross-site integration architecture (the system-of-systems that is formed between the individual San Diego County detentions centers). For this integration architecture, we had several architectural documents available. Of specific interest were the parts of those documents that described the issues encountered when applying an eventually selected technical solution. Namely, the integration was implemented using Oracle's middleware solution that provides a peer-to-peer solution for data consistency. Using this solution, when some part of the detainee data is modified, this information is propagated to other JIMS sites in a peer-to-peer manner (the specific algorithm used is called *n*-way multi-master replication).

At the time of JIMS development, Oracle's technology was tested only on systems in which the consistency had to be maintained across 3 sites. Hence, once the designers started using and testing the planned 7-way master replication, they encountered significant data consistency and system performance problems. Based on further system-of-systems simulations, the designers stated that a better solution would have been using the previously tested 3-way multi-master replication, while the remaining 4 JIMS sites would operate as clients to the three master sites. Due to the wrong choice, the integration took more time to develop and was less stable than desired.

The main idea when applying the integration styles matrix is to determine the appropriate candidate styles for a desired integrated architecture based on the required non-functional properties. Below, we describe how the integration styles matrix could have been used to circumvent the problems encountered in JIMS cross-site integration and arrive at a better hybrid solution.

For the JIMS cross-site architecture, we determined eight crucial non-functional requirements. Table 5 depicts the rows of the integration styles matrix cells for the properties of interest. The lower part of Table 5 contains summaries of the impact that selection of a particular integration style element would have on the desired integration quality. This summary simplifies the elimination of unnecessary elements. For example, the summary suggests that the specific integration at hand does not need adapters and translators; note that this is consistent with the intuition that integrating homogeneous systems should not involve data and protocol translation. Similarly, the summary suggests that hub and spoke is a bad solution for integrating the seven JIMS sites due to, e.g., required security, robustness, and reliability. Further analysis of the topology dimension of an integration style discovered that a pure peer-to-peer solution, which was used with limited success in JIMS implementation, has potential data consistency and distributed transaction problems. Hence, using the matrix would have brought these potential issues to the forefront of the design process.

While this analysis discovered certain limitations of a peer-to-peer solution, we note that both point-to-point and shared bus topologies have a similar overall score and are not clearly superior to peer-to-peer. However, by analyzing specific problematic matrix rows, we arrive at a conclusion that using a hybrid approach –a combination of a peer-to-peer and shared bus solution– would circumvent the individual deficiencies of both. Notably, this solution was retrospectively seen as the most appropriate solution by the actual JIMS designers.

We performed similar analysis for the linkage dimension of an integration style. The specific solutions we reached were that messaging was the most effective linkage mechanism to the problem at hand, which is consistent with the way distributed database transactions are typically done. Furthermore, assessment of the specific deficiencies of messaging discovered that in addition to discrete messages, the integrated systems should connect via data streaming connectors whenever high amounts of data need to be transferred across the JIMS sites.

| Integration styles vs. Properties | | Topology | | | | Linkage | | | | Connector | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Point-to-Pt. | Hub-Spoke | Shared Bus | Peer-to-Peer | Shared Data | Mess. | Explicit invoc. | Data Stream. | Adapt. | Transl. | Arbitr. | Distr. |
| Interaction | Distributed | o | + | + | + | o | + | + | + | o | o | + | + |
| | Secure | + | - | o | +/- | - | o | o | o | o | o | + | - |
| | Data intensive | + | - | - | + | + | - | o | + | o | - | + | + |
| | Data consistency | o | + | o | - | + | o | o | - | o | o | + | o |
| | Reliable | + | - | + | + | - | + | + | o | o | o | + | o |
| | Real time | + | - | +/- | - | + | - | + | + | o | o | + | o |
| System | Robustness | - | - | + | + | - | + | +/- | - | o | o | + | + |
| | Distributed transactions | - | + | + | +/- | + | + | + | o | o | o | + | + |
| Overall | Positive (+) | 4 | 3 | 4 | 4 | 4 | 4 | 4 | 3 | 0 | 0 | 8 | 4 |
| | Neutral (o) | 2 | 0 | 2 | 0 | 1 | 2 | 3 | 3 | 8 | 7 | 0 | 3 |
| | Negative (-) | 2 | 5 | 1 | 2 | 3 | 2 | 0 | 2 | 0 | 1 | 0 | 1 |
| | Conditional (+/-) | 0 | 0 | 1 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

**Table 5: Integration Styles Table for JIMS Cross-Site Architecture**

To summarize, applying the integration styles matrix to gain further design insight has proven successful as it helped to (1) quickly "drill down" on a small set of potentially beneficial design options, (2) reuse existing expert knowledge, (3) identify potential issues early in the design process, and (4) identify better alternative solutions. Furthermore, the JIMS case studies performed for multiple integration levels (cross site integration described here vs. integration with external systems omitted for brevity) have demonstrated the sensitivity of the matrix – the matrix has helped to arrive at distinct and relevant solutions for the different integration problems. Finally, the documented problems of the JIMS system have helped us discover and document additional rationale related to the relationships that appear in the matrix.

## 7.7 Social Media Extensions

Recent work has produced a prototype version of iCBSP that utilizes a social media framework for the interaction among stakeholders. This builds on the current social media platform being utilized for the Win-Win methodology (Winbook). This work is experimental, but is expected to produce more effective and rapid stakeholder exchanges in requirements reconciliation.

# 8 SOCIO DECISION-MAKING IN NET-CENTRIC REQUIREMENTS MANAGEMENT

Requirements management in a net-centric enterprise is a complex socio-technical problem. Technical aspects have been addressed extensively in the previous two sections. This section utilizes a case study to illustrate and study socio aspects of this problem, since the socio aspects often are more difficult to address than the technical ones. The case study selected is the System Design and Development (SDD) phase of the F-35 Joint Strike Fighter (JSF) program conducted by Lockheed Martin Aeronautics (LM Aero), Northrop Grumman (NG) and BAE Systems.

## 8.1 MOTIVATION, BACKGROUND AND KEY QUESTIONS

The net-centric enterprise is increasingly advocated as a means for addressing complex problems by organizations. In many instances, the previous model of vertical integration is not adequate to address the complex challenges of today's environment, as a single firm or agency may not have all the capabilities needed. This is true of both government and private industry.

The transition from the model of vertical integration, with its emphasis on command-and-control, to the model of the net-centric enterprise, with its emphasis on collaboration and negotiation, is transformative (i.e., a major change as opposed to incremental improvement). Thus, this section uses a framework of enterprise transformation (Rouse, 2006) to study decision-making and non-technical aspects associated with a capability and requirements management effort in a net-centric enterprise. This framework considers the following points as underlying how transformation occurs.

- Value deficiencies drive enterprise transformation. Often, a value deficiency is a capability that is needed, but not yet achieved.
- Transformation is enabled by changes to work processes. In SDD, the changes primarily consisted of the processes needed to support the collaboration between partner organizations and other stakeholders, both internal and external.
- Transformation must address the successful allocation (re-allocation) of attention and resources.
- Management decision-making is critical to successful transformation. This includes leadership, strategy and problem-solving.
- Transformation occurs in the context of social networks and organizational culture.

We argue that the enterprise transformation framework is an appropriate and useful approach to studying the socio aspects of requirements management in a net-centric enterprise because of the focus on the above factors.

This study focuses on the decision-making processes and methods involved in managing the integrated capability development to support the SDD phase of the F-35 JSF program. The F-35 program operates in a net-centric enterprise consisting of multiple organizations under a common umbrella that operate semi-autonomously. This enterprise consists of three partners, LM Aero, NG and BAE Systems, plus a number of major suppliers, multiple U.S. military services and an international set of customers. Within LM Aero, a number of organizations had to collaborate, as well. These included the Information Systems & Technology (IS&T) organization, the F-35 program organization and the various functional organizations aside from IS&T (e.g., Engineering, Production Operations, Supply Chain Management, etc.). Since the focus was on lifecycle design, one intent was for the functional organizations to collaborate to provide a set of enterprise solutions that worked seamlessly over the entire program lifecycle (as well as future program lifecycles). This was a new way of doing business for these organizations, as opposed to the traditional approach of using vertical integration to design and produce such aircraft.

This collaboration required a substantial new set of technical capabilities in terms of IT systems to support the collaborative design activities. These capabilities had to be decomposed into requirements and assigned to particular systems within the integration. One key technical capability intent for the F-35 was "digital from design … to flight line and support." We particularly focus on that capability in this case study. This capability development was to be accomplished under significant time pressure within the transformation process and was strongly affected by legacy functions, processes and systems. In addition, there were substantial non-technical issues. For instance, the three partner firms were collaborators on this program, but also competitors in other areas. Thus, there was resistance to the idea of sharing particular capability details that were proprietary in nature. Also, while a common vision could be expressed for how the program should operate, there were many approaches to achieving that vision, and each of the collaborators might favor different ones. Finally, in any major organizational change effort, there is resistance to change via reversion back to previous ways of doing business. Thus, one of the most difficult tasks is to keep stakeholders aligned in a transformation effort.

The questions addressed in the case study are the following:

- What is the role of company executive leadership in setting the operating framework for this type of capability development in a net-centric enterprise?
- What is the role of the technical management in developing these capabilities?
- What are the challenges that both groups must address?

- What changes must be made to the way business is done?
- What constraints affect the capability development?
- How is the initial alignment set among the heterogeneous stakeholders in the net-centric enterprise, and how is it maintained over time?
- How are capability intents decomposed into requirements?
- What types of decision support tools and knowledge management tools are used?
- What would the leadership and technical management groups have done differently?

## 8.2 Case Study Approach

Our net-centric enterprise case study topic is focused on the global enterprise product design challenge of the JSF. Specifically, we examined the design, deployment and use of the JSF global IT strategy that now enables the JSF program management, product lifecycle realization (including the lifecycle digital thread), and system performance validation.

We engaged both the JSF program leadership and the LM Aero leadership to discuss their willingness to provide the information needed for a successful case study. Based on these discussions and our literature review of the JSF vision, we knew that the JSF program needed outcomes from the IT strategy would be transformative and not achievable by existing legacy company design and development practices.

Figure 14 provides the Tennenbaum Institute framework (Bennett, et al., in press) for understanding transformations such as "global enterprise digital design of the F-35 military aircraft." We organized our case study approach around two themes from Figure 14:

1. The transformation processes (leadership, capabilities and delivering solutions) provide the understanding of the transformative intents of the outcomes, the requirements for the capabilities, and the specific tools needed to deliver the intended new solutions.
2. The stages of transformation risk identify the transformation stages where focused risk mitigation provides a successful outcome for the global enterprise transformation. In particular, the executive leadership of a global enterprise must continually focus on stage I to maintain agreements, alignments and commitments.

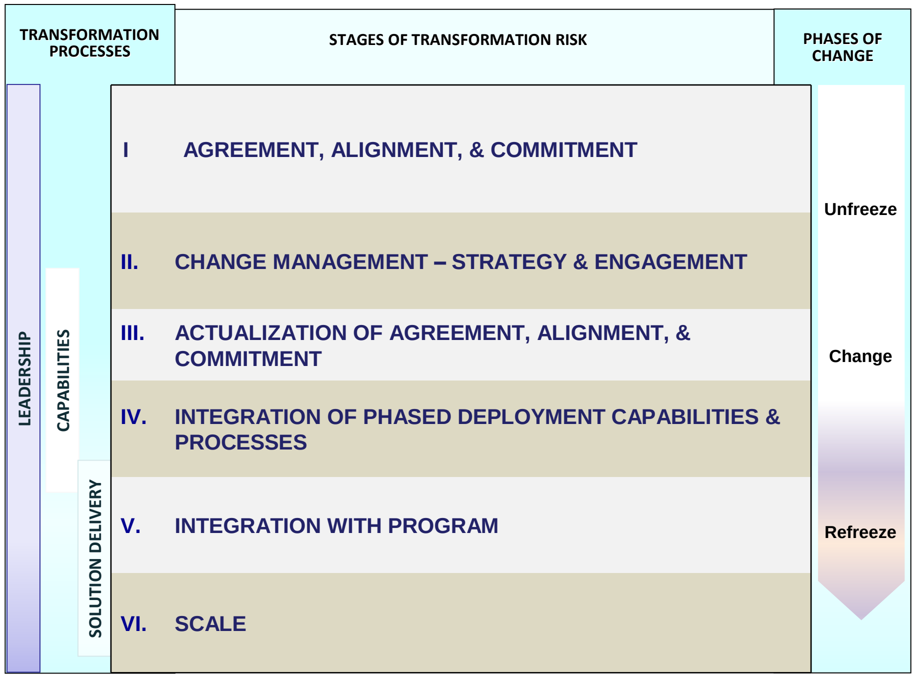


**Figure 14: Framework for Understanding Transformation**

Next we constructed the appropriate questionnaires – one for leadership (2 Topics and 25 Questions) and one for technical managers (3 Topics and 40 Questions); the topics and questions were crafted to allow determination if the stages of transformation risk were addressed and if alignment of intents/requirements were effectively maintained within the enterprise transformation processes.

Knowing that enterprise stakeholder alignment is a key to successful transformation we requested that LM Aero target the questionnaires on the following "JSF stakeholder ripples" within the JSF enterprise: the core IT transformation team, the implementation stakeholders, the JSF stakeholders and the LM Aero company stakeholders (Figure 15). Since this JSF SDD Phase for the global digital design has been completed, we also requested that LM Aero obtain questionnaire responses from past and current employees who were involved in this phase of JSF during the period of 1998-2010. We did not have the resources, primarily the time, to include stakeholders from ripples 4-6. Including their responses is an item of future work.

Figure 15 also outlines the general elements of our case study approach – focus, scope, input, and analysis.

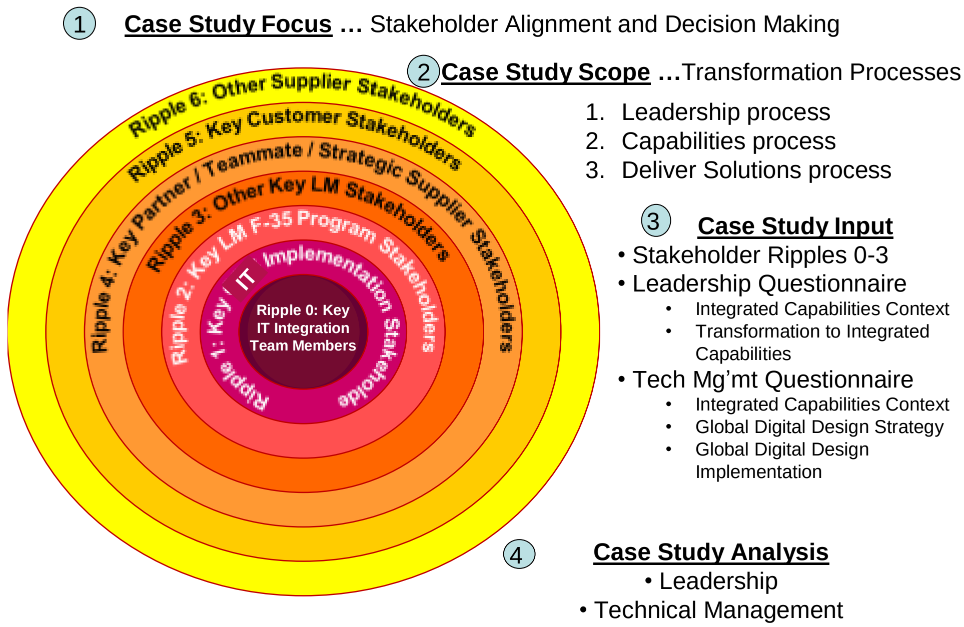


**Figure 15: Summary of Ripples and Case Study Approach**

The two questionnaires (leadership and technical management) that provided the input for our case study analysis are provided in Appendix B.

## 8.3 Summary of Leadership Responses

The leadership questionnaires (2 topics and 25 questions) were completed by JSF SDD leadership stakeholders representing ripples 0-3 and returned to the researchers. The responses were in-depth and informative. Each set of responses for a topic/question was analyzed by the researchers from the perspectives of alignment and decision making as well as any "take away or lesson." A few of the most relevant topic/question analysis follow.

### 8.3.1 Leadership Challenges

Take Away: JSF SDD management and operating challenges were transformative and wide ranging. The leadership team, due to its inability simply to utilize a legacy approach, lacked an experience-base to draw on for addressing many of the transformative challenges; however, the leadership had a solid understanding of what the challenges were and had developed enterprise (LM-NG-BAE) leadership alignment on how to proceed via their pre-contract award team engagements and discussions.

Summary of Responses: The questions for the top leadership did not focus specifically on the IT digital design challenges but were intended to capture the executive level perspectives on overall JSF challenges and to observe where IT challenges fit within the mix. We asked the leaders to identify their perspective of the top 4-5 challenges they faced at the beginning of the SDD phase. The JSF SDD challenges identified ranged from "writing 5 million lines of code" to "operating as one integrated Program team instead of past approaches of having three separate team mates each with their own piece of the Program." Challenges also included the establishment of a JSF Program Integrated Master Schedule (IMS), dealing with intellectual property rights of the three partners, not having key propulsion contractors as part of the JSF Program Team in Fort Worth, TX, and reporting to one specific government agency, the JSF Program Office (JSFPO), while being required to deal with all services, domestic and international, as customers.

With all these "first ever" challenges faced by the JSF Program, it was also clear that the global enterprise IT system for SDD was also a challenge and high priority; in fact leadership noted that "new practices, systems, and infrastructure were needed to enable the JSF team to operate as designed." The specific IT related challenges noted by leadership were the timely documentation of decisions made by disparate organizations and individuals, electronic data repository, ability of the full design team to operate in a set of shared systems, and a JSF-wide product data management system

### 8.3.2 Does SDD Change "How Business Is Done"?

Take Away: The "to-be" operating capabilities of the JSF SDD phase were transformative and required changes to how business and work had been conducted.

Leading and conducting a transformation is dramatically different than leading continuous improvement or incremental change. Operating in an enterprise is transformational and demands continuous leadership engagement for success.

Summary of Responses: All responses from leadership indicated the answer was, "yes, both JSF and SDD change the way business is done." Examples follow:

- Structure: A tightly integrated teaming and management arrangement between LM-NG-BAE would require the JSF Program Team (located at LM Aero in Fort Worth, TX) to operate as a global enterprise that, in many respects, would stand apart from LM Aero. For example, JSF Integrated Product Teams (IPTs) had representatives from all three partners and the lead could be from any of the partners.
- Processes: As an example, the JSF teaming approach required changes to LM Aero Human Resources (HR) processes to support keeping rosters on the Program as though they were all from a single company. Additionally, all

designers from each company had to be capable of releasing International Traffic in Arms Regulations (ITAR) data so that when a design was stored it was available to all international participants who had authority to access. This process and training was a major departure of past processes that had a single and separate release authority.

- Systems: During SDD the JSF Program required new global enterprise systems capabilities for managing the program, realizing the product, and for validation/simulation of product and performance. These systems had to incorporate global access of all designers and security.
- Skills: The JSF Program needed a high number of enterprise leaders due to the global nature and management complexity of the Program.
- Staffing: LM Aero had implemented a major overhead cost reduction initiative prior to winning the JSF Program. Functional staffs had been reduced and much of the functional talent and leadership had migrated to other important LM Aero Programs (F-16, C-130J and the F-22). JSF Program startup faced a huge staffing and functional leadership challenge.
- Culture: Any global enterprise will have cultural challenges and JSF faced a myriad of these challenges brought on by cultural differences between countries, between public and private entities, between partner companies, between companies and suppliers, between company functional organizations, etc.

### 8.3.3 Leadership's view of SDD "to-be" capabilities

Take Away: Leadership had a clear view of the "to-be" capability state and that state was transformative. These capability requirements were flowed to the technical management for action.

It is unclear if the public sector Department of Defense (DoD) stakeholders understood or cared about their needed roles and responsibilities in achieving overall enterprise capability success. The roles and responsibilities of public sector stakeholders in the design and operation of an enterprise remains a significant uncertainty.

Summary of Responses:

- Managing the Program: 1) provide timely identification of issues and a structure for resolution, 2) implement IPT structure with members from all 3 partners, 3) establish common requirements management system, a single data management system, a common document management system, and the common infrastructure to enable team-wide access to data.
- Design and Develop F-35: 1) provide real-time access to all designers and managers within the security and management constraints, 2) enable appropriate designers and managers to make and document trades offs, and 3) develop software and control the weight.

- Establish Global Team and Concept of Operations: 1) genuine desire to become a global operation and enable appropriate individuals and companies to make and document decisions, and 2) convince team mates that there were no offsets and work was going to "best value."
- Create enabling skills, methods, tools, and systems: 1) keep focus on developing needed capabilities as soon as possible, 2) use best practices from partners, concentrate on core competencies, and 3) hire outside expertise for rest.

### 8.3.4 Constraints

Take Away: Many policy and contract constraints were known but many more were "discovered" as the program matured. Complex programs such as the JSF have an element of "You don't know what you don't know."

Summary of Responses: The contract was awarded in October 2001 soon after the 9/11 attack. The DoD kick-started the program launch the day after the award which took away the originally scheduled 90 day to get plans and concept of operation aligned.

Inertia-to-change within partner companies often constrained the ability to enable the JSF IPT enterprise structure and to embrace the needed business changes to replace legacy practices, tools, methods, and processes.

Some constraints emerged as the program proceeded. For example, corporate networks and firewalls were designed to protect "corporate crown jewels," but the JSF Program was going to create "crown jewels" that would be outside the corporate firewall. Solutions had to be invented to work around this real policy issue. There were continuing constraints due to issues between the military service focus on performance and the system commands focus to control costs.

### 8.3.5 Alignment of Stakeholders

Take Away: Establishing an enterprise approach on a program such as JSF where there are three separate corporations involved may result in a common and supported vision for the effort but will also result in widely varying approaches for achieving the vision. JSF SDD leadership experienced this alignment divergence under the time pressure of launching the program. The convergence methods they used were conflict resolution, formal negotiations, and "work around the problem." Decisions were electronically captured and distributed.

Leadership responsibility to maintain alignment and commitment throughout the enterprise and over the program lifecycle is essential for a successful transformation.

Summary of Responses: The vision was well understood but the approaches for achieving it varied widely. The F-35 team believed from the beginning that new systems and practices were going to be needed. Many overlapping and conflicting ideas were brought in from executives based on their personal background, marketing from software venders, and perceptions of what the customer wanted.

Sometimes there were differences in methods of operation or capabilities that presented challenges in interfacing with each other. Many of these could be resolved by identifying changes to the infrastructure that would enable the collaboration to proceed; other times the changes/capability improvements were onerous (costly or just “we don’t do things that way”) to one party or the other and was resisted. These had to have negotiated settlements that took time, energy and money.

Other issues were simply a difference of opinion on what was needed or what should be done. Sometimes these were readily identified and a negotiated settlement reached. Other times the differences were either not obvious or were purposely hidden - these were much harder to deal with. In some instances, security constraints made it virtually impossible to reveal the true nature of why something needed to done in a certain manner and it made no sense to the “uninformed” party. It would therefore be very difficult to get buy-in.

A formal negotiation for conflict resolution was a part of gaining alignment. The process worked well where there was solid alignment between stakeholders on the solution. It did not work well where some stakeholders did not like the solutions identified and continued to “fight” the solution or develop their own solution “under the table.”

### 8.3.6 Was F-35 SDD Successful in Terms of SDD Intent – What Would You Do Differently?

Take Away: JSF SDD was successful from the perspective of creating the global infrastructure for design collaboration. JSF SDD had some shortfalls related to program design/weight (this is not an IT issue but it is a decision making and alignment issue worthy of analysis) that adversely affects the production and sustainment phases.

Some do-over’s included balancing the functional organizations and the program, management of infrastructure projects, specific design and people decisions, and the public sector engagement/role.

Summary of Responses -- What would you do differently?

- Employed more/better capability in functional organizations earlier as opposed to deploying many of their best people to the program.

- Better enforcement of stopping unique infrastructure development by the program.
- Earlier identification and development of the infrastructure changes needed by the program.
- Created an independent team to determine why the government was predicting a higher weight growth than JSF engineers.
- Not much – the planning was extensive – and really set the stage for success – but in some cases the extensive planning was overwhelmed by the complexity and extreme pace of the program

## 8.4 Summary of Technical Management Responses

The technical management questionnaires (3 topics and 40 questions) were completed by JSF SDD technical management stakeholders representing ripples 0-3 and returned to the researchers. It should be noted that IS&T led the IT integration team (ripple 0), whose members were deployed to work with another functional organization or the program. There they were responsible for consolidating and coordinated their specific organization's requirements both internally within the function and across the enterprise via the other IT integration team members to develop the integrated IT solution to allow the F-35 Program to operate successfully.

Similar to the leadership responses, the technical management responses were in-depth and informative. Each set of responses for a topic/question was analyzed by the researchers from the perspectives of alignment and decision making as well as any "take away or lesson." A few of the most relevant topic/question analysis follow.

### 8.4.1 Technical Management Challenges

Take Away: The JSF SDD IT development and operation were transformative and wide ranging -- help manage the global program, enable the global enterprise, integrate capabilities to design and build and support the products, and assure IS&T global enterprise skills, methods, tools, and system.

Summary of Responses: LM Aero had a great deal of technical capability that could be leveraged. Its main challenge was to develop the processes that would support effective use of the technologies and systems by multiple stakeholders and collaborators in meeting JSF program needs, as well as constraints. These processes were impacted by the global nature of the program (across many time zones, cultures and languages).

The leadership had recognized that IT was on the critical path for JSF management and product development, which led to the launch by the LM Aero IS&T functional organization of a significant pre-contract award item, "virtual product data initiatives

(VPDI)," to prepare for the IS&T role in enabling the management and operating capability of the JSF global enterprise system. This provided a platform that the IS&T organization and later the IT integration team could use to meet the challenges.

Still, there were considerable technical hurdles to be met, including providing real-time access to design data from globally dispersed sites, abiding by ITAR, and ensuring continued alignment among stakeholders at the level of capability development and the various approaches that were alternatives.

### 8.4.2 Existing and Needed Capabilities

Take Away: Most of the component technologies were in place to support the SDD phase of JSF. These included applications (e.g., requirements tools, design and analysis tools, middleware), as well as information system architectures and technologies (e.g., Wide-Area Networking (WAN), virtualization, etc.). Once again, the VPDI initiative was critical in setting up the needed technical capabilities. There was a good understanding of the technical capabilities needed; however, what was missing across the enterprise was the integrated application and data availability, as well as effective processes to support the collaborative design capability. The process capabilities were not as well understood up-front and had to be "discovered" as the program progressed.

Summary of Responses:

Existing capabilities leveraged:

- LM Aero had extensive experience with traditional applications for requirements, design and analysis and middleware. In addition, it had experience with architectures and technologies that could be used to support the collaboration, such as WAN and virtualization.
- The VPDI effort enabled a fast-start to new capabilities.
- LM Aero had some limited experience sharing data with partners (F-22 program with Boeing). Both BAE and NG also had some experience with partner-based collaborations and data-sharing.
- The Product Data Management (PDM) effort had been initiated in the VPDI, and one of the partners had extensive experience with PDM.
- Bill of Materials (BOM) control system had been developed in the F-22 program.
- Most partners used a common Computer-Aided Design (CAD) application.

Capability needs included:

- An efficient front to back process was needed that would tie the team members together to support the manufacturing floor and supply chain needs under the planned high production rate.
- Commercially available network encryption equipment was needed.

- A common and accessible PDM application and associated infrastructure for 3D product data was needed. A process that accounted for the new design responsibility breakdown had to be designed. For this breakdown to work, the system needed to support sharing of data in a near real-time mode and needed role-based controls to support ITAR requirements.
- Tools were needed to provide alignment of design concepts to support future manufacturing and sustainment.
- Capability was needed for sign-off on design decisions to be performed entirely with digital data.
- Coordination was needed of business processes among partner organizations to behave as one process.
- Ability was needed among partner organizations to overcome rules in the partner organizations to share data.
- Ability was required to handle the collaboration with minimal additional overhead.
- All the partner organizations and major suppliers needed their systems set up to plug into the overall system. This would involve substantial work.

### 8.4.3 What Business Changes Were Needed?

Take Away: Due to the scope and scale of the transformative approach to design, a variety of changes were required. Change management processes had to account for external stakeholders. A variety of support teams and coordination mechanisms had to be stood up to address numerous issues ranging from the global nature of the collaboration to software licensing. Division of work and oversight among multiple organizations and firms was a major challenge. Culture change issues included data ownership norms, common processes among different organizations, and tendency to revert to previous, known processes. These drove significant business changes, some of which occurred in mid-stream.

Summary of Responses:

- The change management process had to account for stakeholders external to LM Aero who would be impacted by changes that groups within LM Aero might make.
- A dedicated infrastructure support team was needed to support the global nature of the infrastructure. Technical groups were set up to support the architecture and infrastructure design and implementation.
- The multi-company nature of the collaboration and the shared tools caused licensing issues with software vendors that had to be resolved.

- Development was funded by the LM Aero IS&T organization, but typically managed by Engineering. Sometimes this caused more focus on the technical aspects of the software development and less on cost/schedule aspects.
- Data ownership issues had to be resolved between users, who felt that if their data was not in their tool, they had ceded ownership. This was addressed by adjusting the PDM system to ensure that users had a better understanding of the workflow. It also resulted in a more flexible process that facilitated implementation of changes when needed.
- There had to be conscious effort applied to prevent reversion by users to the legacy systems with which they were familiar, since those systems could not support the program's needs.
- The workflow and data marking had to be designed to support the collaboration yet also adhere to ITAR requirements.

### 8.4.4 Constraints

Take Away: Constraints were imposed on a number of fronts. Availability of enough skilled people was an issue. ITAR was cited as a key constraint that could not be violated. The time zone and global geographic differences imposed constraints on processes technical performance that had to be overcome. Cost and schedule were, of course, constraints.

Summary of Responses:

- There were not enough skilled project managers, and sometimes project management issues delayed capabilities.
- Sometimes, there were not enough technical Subject Matter Experts (SMEs) available to support the domain expertise due to support of other programs or due to other responsibilities.
- Key talent was available at higher management levels. Additional talent had to be hired or sourced via vendors.
- The VPDI effort mitigated schedule constraints that otherwise would have been problematic.
- ITAR, time zone and geographic difference constraints could be addressed in a technical sense, but the cultural issues associated with operating a global program were more difficult.

### 8.4.5 What Was Needed from the Leadership?

Take Away: The technical management felt that the leadership needed to participate in the effort actively, provide public support, and champion the need for transformation. The leadership needed to engage in a top-down manner and ensure that the technical teams did not feel abandoned. They also needed to understand that the distributed nature of the system would have significant impacts on schedule and cost.

### 8.4.6 Technical Management Inter-Organization Strategy

Take Away: The strategy was to use a common set of tools (PDM, CAD) across the set of firms and organizations involved in the design with access to a single engineering BOM and manufacturing BOM. This was led by the LM Aero IS&T organization, which worked with the IT organizations at partner/supplier sites to implement the necessary systems. There was general understanding of the vision and sense of urgency for the SDD capability intents.

Summary of Responses:

- The IS&T organization was to support these intents via the WAN, the Virtual Processing Center (VPC) data store, and the overall IT architecture. The functional organizations were to provide integrated tools and processes that had not been available before.
- Multi-company IT teams were used to coordinate system and infrastructure integration. The IT integration team (ripple 0) provided leadership and macro-standards, while each partner was responsible for its own implementations.
- The IT integration team worked closely with the program and engineering organization to extract needed requirements.
- SMEs were involved from a variety of domains to ensure that input did not consist solely of designer needs.
- The main toolset was determined prior to engagement with partners. However, their experience and information was to be incorporated into the solution.
- One part of the strategy was to link the master BOM to partner Enterprise Resource Planning (ERP) and Manufacturing Resource Planning (MRP) systems to eliminate Purchase Order (PO) traffic in production. However, this was not pursued due to later funding decisions.

### 8.4.7 Maintaining Alignment and Addressing Conflicts

Take Away: Teaming structures were defined and adhered to between the various organizations involved – LM Aero IS&T, F-35 Program, LM Aero functions and partner companies. Conflicts were identified and resolved mostly via traditional means (i.e., no decision support tools).

Summary of Responses: In the program, there was a unit devoted to having the program remained aligned with the vision (later removed due to budget reductions). The IS&T organization led the IT integration team and remained aligned by significant planned interaction and communication with the program organization (e.g., senior IS&T personnel assigned to the program, regular meetings, discussion of scope/schedule/performance, etc.). Functional areas worked with the program organization to stay true to program needs as they evolved. Functional organizations at first were not aligned, but became so later due to leadership influence and importance of the program.

There was little in the way of formal negotiation methods or models used to identify and resolve conflicts. The program organization utilized a BOM team to identify and manage conflicts. This extended up to the program-level Change Board which approved all BOM policy. This process did not extend to program-functional organizations. F-35 was not the driving program for LM Aero in the early stages of the SDD phase. Thus, some functional organizations often fell back to the F-16 way of doing business, resulting in conflicts. In the IS&T organization, many conflicts were addressed in an ad-hoc manner. Often, technical conflicts were resolved by IT personnel, while process conflicts were resolved at a higher level (e.g., IPT lead).

General IT information was shared openly. Some systems, though, required role and user based access control due to proprietary or sensitive data. The collaboration did create issues contributing to increased cost/schedule. For instance, there was a trade-off between delays caused by the increased number of approvers needed upstream in change control management due to the number of stakeholders versus potential downstream costs/delays associated with making changes that have unforeseen negative consequences due to lack of consultation with the right stakeholder. Also, there was lack of alignment on the importance of adhering to a common version of CAD software. The cost/schedule impact was not appreciated when one of the partners upgraded their version.

### 8.4.8 Decomposing Capabilities to Requirements and Reporting Progress

Take Away: The high level vision for capabilities provided a solid foundation. The technical management group and employee teams were able to decompose these to business process requirements and system requirements, although this was not done in a formal manner. This was an iterative process. There was concern that some decisions were made without a global perspective, causing conflicts later in the process. Rationale for decompositions and resulting requirements was done reasonably. Much of the process would be repeatable.

Summary of Responses:

- While the decomposition process was not done formally, many of the team members had been involved in a previous effort that provided an experience base for this kind of effort. Also, results were reviewed with stakeholders at key intervals to obtain approval or redirection.

- One issue was that once the tools development transitioned from the VPDI to various functional centers, there did not seem to be an overall strategy that linked the various development efforts.
- The process unfolded much like a rapid prototyping approach with SMEs who were familiar with the needs of their functional center adapted to operate in a global program. Then it moved into a pilot with users who tested the capabilities with increasing levels of fidelity.
- Progress reporting was done largely by traditional means – review meetings, tools such as Microsoft Project, risk identification and resolution, and change board processes.

### 8.4.9 Decision Support Tools and Knowledge Capture

Take Away: For the most part, there were no decision support tools used in developing the integrated IT for digital design of the F-35. There was no systematic approach to capturing knowledge from the integration process to support other future programs, despite an intent that such knowledge capture should happen.

Summary of Responses: There was limited use of decision support tools such as value based requirement analysis and quality function deployment. PDM tools were evaluated using an evaluation/decision matrix. Most decisions were framed around meetings and kaizen-type events.

While knowledge capture was not systematically pursued, many of the processes, procedures, methods and templates were documented for potential future use. It is not clear whether this will facilitate actual future use, though. There were concerns about limits on potential reusability that related to the unique scale/scope of the JSF program, lack of personnel that could be deployed away from JSF to other programs, differences in cultures between LM Aero sites, and the (incorrect) perception by the leadership early-on that the integration was a failure. As tools mature, there is an intent to develop an open architecture accessible by other programs.

### 8.4.10 Was F-35 SDD Successful in Terms of SDD Digital Design Intent – What Would You Do Differently?

Take Away: There are a variety of ideas for doing things differently. For the most part, the effort was viewed as being successful, so these ideas are geared to improving the success. Process aspects of the SDD tended to have more suggestions for improvement over the technical aspects.

Summary of Responses: The importance of the leadership providing top-down support and reinforcement for the need to change was emphasized, as well as the need to push this out to the partner companies.

The process should have included a better up-front understanding among the enterprise stakeholders of what the digital design concept entailed. There was a feeling that most of the focus was on the tools and systems and not on the processes or behavioral changes that needed to occur. Other examples for improved processes include better communication among stakeholders in designing the integration, better understanding of the concept of operations for the integration, and standing up an organization specifically devoted to the development and integration to prevent functional silos from allowing development/integration to drift from the common vision. As previously noted, the IS&T organization established the IT integration team by deploying members to work specifically with the program, LM Aero functional organizations, and the partners on the challenge of a global enterprise digital design.

Examples for improved technical aspects of SDD include a common toolset across LM Aero, more attention to data standardization and synchronization, uniform adoption of CAD tools by all organizations, and uniform and shareable design data levels between partners to facilitate changes in design authority

## 8.5 Overall Lessons Learned and Future Research

The following points represent important lessons learned from the case study, generalized to observations that likely apply to many/most requirements management efforts in net-centric enterprises.

- The overall SDD effort was transformative. It required numerous changes to the traditional way of doing business both at the executive leadership level and at the technical management level. The transition from traditional vertically integrated organizations to a net-centric enterprise is a transformation, and it has important implications for how capability and requirements management can be conducted successfully.
- Leadership engagement was critical throughout the SDD. As with any transformation effort, leadership engagement throughout the effort and extended out to partner organizations is vital.
- Maintaining alignment by stakeholders across the enterprise is a vital but very difficult task. In SDD, there was a common vision for how the program and net-centric enterprise should work, including the capability development effort. However, there were many different approaches across the set of stakeholders. In addition, moving from enterprise leadership into specific organizations, there is an increasing tendency to revert to legacy processes and systems without sustained executive leadership support of the transformation. Using such legacy

systems and processes would not have met SDD intents. Similarly, it is likely that in general capability intents in net-centric enterprises will not be met via legacy approaches.

- The VPDI effort was a major enabler of success. Setting it up early was a cost risk, but provided early capabilities, setting the stage for progress under schedule constraints. An early initiative like this for any capability and requirements management effort should be considered, along with its potential risks.
- Teaming efforts among organizations were effective. Some want a more formal organization specifically set up to handle the effort. In any capabilities and requirements management effort across a collaborative enterprise, effective teaming approaches are critical to success. No teaming structure will be perfect, so effective teaming should be the goal.
- Many constraints were not known at the beginning of the SDD, but rather had to be "discovered." In net-centric capability and requirements management, it may be advisable to conduct an active constraint/problem discovery process.
- Tools and technologies were not necessarily easy to address, but they were doable. Processes and behavioral changes, on the other hand, were more difficult and required ongoing work and adjustments. Up-front thought and study of the likely impediments and solutions to effective processes is an important foundation for success in these types of efforts.
- Conflicts among stakeholders were resolved largely by traditional means in SDD. This seemed to work, for the most part, although there were instances where decisions without global perspective caused unanticipated conflicts, and there were conflicts in between the public and private sector organizations. There is potential value in research aimed at structured processes for conflict identification and resolution, especially as applied to public-private partnerships.
- There was intent for systematic knowledge capture in SDD. Some knowledge capture did occur through process documentation, but there are concerns that knowledge capture for future use did not reach potential. This is an issue in general. Often, there simply is not time to do the work needed and then store the approach for future retrieval, largely because existing tools are not adequate. Developing effective knowledge capture and management approaches is an avenue of future research.
- There were many successes in the decision-making methods used in the F-35 SDD. There were also suggestions on the decision-making effort's shortcoming and how it could have been improved. These provide important lessons for other similar efforts involving capability and requirements management in net-centric enterprises.
- It should be stated that the private sector organizations in the F-35 enterprise were motivated for program success because it impacted their financial performance. Public-sector organizations have other motivations and reconciling these public-private sector partnership intents within a common framework is difficult. This is a topic worthy of future research.

This case study addressed only the SDD phase of the F-35 program, mainly within Lockheed Martin Aeronautics (ripples 0-3), and also focusing on the leadership and technical management groups. Of course, this program is much more extensive and involves a variety of other opportunities to investigate capability and requirements management in a net-centric enterprise. A few potential avenues of future research include the following.

- Specify formalized decision-making processes, decision rights, negotiation methods and tools, and knowledge capture methods and tools for lifecycle capability and requirements management in net-centric enterprises.
- Study the detailed technical requirements-to-architectures issues in SDD via interaction with technical capability development and delivery personnel. What were the key issues, and how were requirements management decisions made at this level? Would the MPTs from this research have aided the detailed IT development work?
- Extend the study to include partner organizations, strategic suppliers and key customers (ripples 4-6). What decision-making issues did the partner firms face in the collaborative effort, and what was their perception of the issues investigated in the current study? What were the key requirements management issues in the global public-private partnership that exists between the program firms and customers, how were they addressed, and which approaches were successful versus less-then successful? How are capabilities and requirements managed effectively in a public-private partnership enterprise?
- Extend the study to include the downstream phases of production and sustainment. Of course, these phases are either on-going or future efforts. Nevertheless, there are issues that can be studied. How did the capabilities and requirements management in developing the integrated IT to support SDD perform relative to the evolution of these two phases and the intents behind them? How did changes in the environment impact that performance?

# 9 VALIDATION

To validate the concepts, approach and MPTs in this research, we conducted surveys, interviews and walk-throughs using subject matter experts with extensive experience in IT integration. In this effort, the focus was on two primary domains – manufacturing and healthcare.

## 9.1 ENTERPRISE IT SURVEY

To initiate the validation effort, a previously developed questionnaire for requirements management in integration efforts was adapted. This questionnaire is used to characterize case studies and is contained in Appendix A of the Phase 1 report of this project (Bodner, et al., 2011). It was placed into survey form, in which respondents were asked the frequency with which they encounter the issue addressed in each question on a five point scale (5 = Very Frequently, 4 = Fairly Often, 3 = Occasionally, 2 = Rarely, 1 = Never).

A group of five consultants were asked to respond to this survey. These consultants work at a firm that specializes in IT integration work for corporate clients with enterprise resource planning (ERP) integration needs (e.g., corporate mergers and consequent needs for IT integration). The frequency ratings were tabulated, and those questions with a median rating greater than or equal to 4 were selected as the basis for a set of in-depth interviews with the respondents. Discussion centered along the lines of (i) what approaches do you use to address these questions now, and (ii) what are the strengths and weaknesses of these approaches. Table 6 summarizes the interview responses.

In addition to the survey questions, other questions addressed which methods and tools were used by the interviewees, and which additional methods and tools would be of interest. Responses are summarized as follows.

- <u>Currently in use</u>: basis experts, MS Excel, MS Project, SAP Solutions Manager, SharePoint, tribal knowledge Visio
- <u>Additions of interest (suggested by interviewees)</u>: data conversion tools, knowledge management, PMI tools, traceability tools
- <u>Additions of interest (suggested by interviewer and accepted by interviewees)</u>: Checklists, game books, templates

| Topic | Responses |
|---|---|
| Intent and Actors | • 80% of interviewees emphasized the great importance of having the right team of business and IT stakeholders<br>• 80% of interviewees emphasized the need for desired business capabilities to drive IT requirements<br>• 60% of interviewees discussed the importance of identifying conflicts among desires and requirements early |
| Decision Making Approaches | • 100% of interviewees emphasized the process of defining priorities<br>  - Options, consequences, tradeoffs, costs, schedule<br>• 100% of interviewees discussed the importance of the "blueprint" and project plan, typically represented in MS Excel and/or MS Project |
| Integration Context | • 80% of interviewees indicated that data conversion is crucial<br>  - Units of measure problems are pervasive<br>• 80% of interviewees noted the idiosyncratic nature of interfaces, both software and user interfaces<br>• 60% of interviewees indicated that legacy issues are common – "It's never just SAP."<br>• 60% of interviewees said that it is essential to understand the business processes to be supported<br>  - Often conflicts across multiple acquired business units |
| Integration Constraints | • 60% of interviewees indicated that the people provided by clients is often a constraint<br>  - Too few business people; non "A" players from IT<br>• 60% of interviews said that legacy constraints are imposed most of the time<br>  - Long-term goal is often elimination, not integration<br>• 40% of interviewees mentioned milestones, schedules and costs, as well as the requirement to work alongside competitors |
| Capabilities and Requirements | • 60% of interviewees discussed priorities in terms of those set in advance, as well as those emerging from discovery and negotiation<br>• 40% of interviewees discussed how priorities are documented and shared<br>• Note that capabilities and requirements were discussed extensively under Intent and Actors |
| Architecture | • 60% of interviewees indicated that architecture issues most often emerged with regard to legacy systems<br>• 40% of interviewees discussed architecture conflicts and the use of middleware, as well as resolution by deciding on winners and losers<br>• 40% of interviewees discussed knowledge sources for understanding conflicts in terms of VS experts or client IT personnel for unfamiliar legacy systems |
| Problems and Exceptions Encountered | • 100% of interviewees said the greatest problem is not having the right people on the team<br>  - "You need an executive sponsor, not from IT."<br>• There were also a range of comments on data incompatibilities, as well as inherent conflicts between SAP and legacies |

**Table 6: Summary Interview Responses**

Based on the interview results, an inferred methodology for requirements management in an IT integration project is the following.

1. Team Formation
2. Blueprinting (SAP Solutions Manager)
3. Project Planning (MS Excel, MS Project)
4. Project Management (Traceability, MS Excel, MS Project)
5. Process Mapping (Visio)
6. Knowledge Management (Tribal Knowledge, SharePoint)
7. Data Conversion (Units of Measure, Translators)
8. Change Management (Stakeholders, Interests/Issues, Competencies)

Several observations can be made. First, there seem to be three critical areas where improved or additional MPTs could make an impact – project planning & management, legacies and data conversion, and knowledge management. Second, the nature of legacies and client preferences determine data conversion issues. The way in which legacies are configured may have an important impact, in addition to the higher-level issue of which legacies are present. Knowledge may or may not be available on why legacies are configured in certain ways, and what the impact is of changing the configuration. Finally, knowledge management is mainly handled via "tribal knowledge" with some SharePoint. Such tools as checklists, templates and game books could help, but they would have to overcome time and cost limitations for creation and maintenance.

## 9.2 NEEDS ASSESSMENT SURVEYS

Next, we conducted a more focused survey to determine needs via responses from subject matter experts in different domains. The particular emphasis was on project planning and management, operational issues, legacies and data conversion, knowledge management and methods and tools.

Survey instruments were developed and deployed to manufacturing-oriented SMEs and to health-oriented SMEs. Note that the manufacturing-oriented SMEs were consultants from the same firm as those who participated in the earlier survey/interviews (i.e., that firm's business is largely focuses on manufacturers). The health-oriented SMEs were from a variety of firms. Seventeen manufacturing-oriented responses were returned, while twenty heal-oriented surveys were returned, with summary results in Table 7.

The metric refers to whether the respondents were asked to indicate their agreement (5 = Strongly Agree, 4 = Agree, 3 = Neutral, 2 = Disagree, 1 = Strongly Disagree) or the frequency with which issues are encountered (5 = Almost Always, 4 = Frequently, 3 = Sometimes, 2 = Rarely, 1 = Almost Never). In the column labeled "MFG," the median response of manufacturing-oriented respondents is noted, while the median response of health-oriented SMEs is noted in the column labeled "HC." A $t$-test was used to determine statistical significance of differences in responses between the two sets of SMEs. Significant differences are noted at the levels of $p < 0.10$ and $p < 0.05$.

| Survey Question | Metric | MFG | HC | *p* |
|---|---|---|---|---|
| **Project Planning and Management** | | | | |
| Projects all begin with formal project planning | Agree | 4.35 | 3.75 | 0.10 |
| Explicit and agreed upon methodology for task prioritization and project execution | Agree | 3.81 | 3.80 | - |
| Reports of project status, percent completion, and projected completion time | Agree | 3.71 | 3.74 | - |
| Methods and tools used to support project planning and management are adequate | Agree | 3.56 | 3.50 | - |
| **Operational Issues** | | | | |
| Diverse stakeholders whose desired business capabilities and IT requirements often conflict | Freq | 3.29 | 3.85 | 0.05 |
| Existing methods and tools for identifying and resolving conflicts are satisfactory | Agree | 3.18 | 2.85 | - |
| Business capabilities are known beforehand and can be decomposed easily into IT requirements | Freq | 3.18 | 3.35 | - |
| Architectural conflicts between component systems resolved by selection from existing alternatives | Freq | 3.88 | 3.53 | - |
| Methods for addressing architectural conflicts are adequate for clients' needs | Agree | 3.82 | 3.35 | 0.05 |
| **Legacies and Data Conversion** | | | | |
| Projects typically have one or more legacies that must continue to operate after integration | Freq | 3.59 | 3.75 | - |
| Projects involve one or more legacies whose data must be integrated | Freq | 4.18 | 3.70 | 0.10 |
| Projects involve discovering data incompatibilities that must be resolved | Freq | 4.24 | 3.47 | 0.05 |
| Methods and tools that support data conversion would be helpful | Agree | 4.12 | 3.90 | - |
| **Knowledge Management** | | | | |
| Valuable knowledge for future projects is gained on legacy issues, data incompatibilities, and related concerns | Agree | 4.35 | 3.95 | 0.05 |
| This knowledge is captured, archived, and shared across personnel | Freq | 3.18 | 3.20 | - |
| Other personnel access this knowledge when they are working on other projects | Freq | 3.18 | 3.10 | - |
| Methods and tools that support knowledge management would be helpful | Agree | 4.12 | 4.10 | - |
| **Methods and Tools** | | | | |
| Your overall approach, methods, and tools could be formalized into a standard methodology | Agree | 3.94 | 3.80 | - |
| Additional methods and tools, beyond those currently used, could make this methodology more powerful | Agree | 4.06 | 4.00 | - |
| A formal methodology with standard methods and tools would help new hires | Agree | 4.12 | 4.05 | - |
| A formal methodology with standard methods and tools would provide a competitive advantage | Agree | 4.00 | 3.95 | - |

**Table 7: SME Survey Results**

A few summary observations are in order.

- Current project planning and management abilities are somewhat mixed across both domains.
- Current abilities to address conflicting requirements and architectures are mixed across both domains.
- Legacies and data conversion issues are generally present, although more in manufacturing than in healthcare IT integration.
- Knowledge management comes across as a real opportunity for improvement in terms of ways to capture, archive and share knowledge across the organization.
- There is general agreement in both domains that their approach and associated methods and tools could be formalized into a methodology, that additional methods and tools would be useful, and that a methodology would benefit new hires and provide a competitive advantage.

## 9.3 SME Walk-Through

The previous validation efforts have addressed the concepts and MPT needs of the user community. We engaged two subject matter experts at Georgia Tech with extensive experience in health IT integration to review the overall methodological framework of this research, as well as the specific MPTs associated with the requirements-to-architecture work. One SME's background was in health IT consulting, while the other's background was in health-related software product development and integrated solutions. The remainder of this section summarizes the feedback obtained from the SMEs.

### 9.3.1 Issues in Health IT

Health IT suffers from a number of challenges caused by the nature of the industry. Interoperability issues arise due to the fragmented nature of the industry, with its heavy emphasis on specialties and the resulting departmental/disciplinary focus of healthcare delivery. Within a healthcare organization such as a hospital, each department wants the best-of-breed software solution for its specialty without regard to optimizing the whole IT system or even providing interoperability between departmental systems. Another issue is that knowledge capture and reuse are not necessarily incentivized, since revenues are based on services provided and billable hours, and margin is not necessarily considered. Finally, there are extensive data compatibility issues across different IT systems, and sometimes even within a particular IT architecture (e.g., a "patient day" is not standardized, since some organizations want it to account for acuity).

### 9.3.2 Feedback on Overall Methodology

The SMEs believed that the overall methodology was reasonable. In particular, they reacted positively to the iterative (spiral) nature of decision-making.

### 9.3.3 Integration Matrix

Feedback on the integration matrix consisted of the following.

- The SMEs resonated with the concept behind the integration matrix, as well as the specific terms used for the row and column labels.
- It was suggested that one be able to apply weightings to the rows to give them priorities. This weighting likely is dependent on the perspective of the stakeholder. Thus, there should be some way also to aggregate weightings across the stakeholder community but still retain a way to see how weightings are influenced by perspective.
- It would be valuable to create an integration matrix at the beginning of a project and then use it throughout. During a project, the tendency is to do work-arounds to address issues that arise. Going back to the integration matrix after a work-around would be important in keeping project discipline (i.e., it could be used as a project management tool).
- Cost should be considered somehow, especially for system properties such as real-time.
- The user should be able to provide his/her own labels for rows. For instance, in health IT, response time is generally more critical than real-time.
- Using the knowledge capture and management feature of the integration matrix would be valuable within a project, as well as across different projects.
- The knowledge capture and reuse would be important in reducing costs.
- The integration matrix could be used to support project reporting and revisiting of previous decisions.
- In terms of the matrix entries, there is a school of thought that says that having an even number of choices is important. Otherwise, the tendency may be to pick the middle value (e.g., pick “3” on a scale of 1-5). Thus, having “+” and “-” as options is good.
- Less experienced software engineers may be more likely to adopt this tool, since they have less of an established way of working.
- One interface suggestion is to change the column labels so that the styles under each style category are more clearly identified as belonging to the category.
- Users need to understand the trade-offs associated with various potential features. Otherwise, they tend to want everything. This tool seems to be able to support that.

- The software engineers and developers need to understand the collective priority on important system properties (integration matrix rows) so that they can create good designs.

### 9.3.4 iCBSP Winbook

Feedback on the iCBSP methodology implemented on Winbook consisted of the following.

- The Winbook platform would be valuable in terms of gathering user input to requirements and features. Having an asynchronous way of doing this is very desirable. The traditional alternative is to bring users in for feedback sessions, which is logistically very difficult.
- One of the SMEs had previous experience with a wiki-based tool that supported user community prioritization of features. The wiki platform had usability problems. A social media approach appears to overcome these problems.
- One idea from this wiki-based tool is to have different categories of users. For instance, the whole user community might be allowed to comment on requirements, features and issues, but only users who have passed some qualification (e.g., committed users) would be allowed to vote.
- Care needs to be taken so that user input is framed in a way that users are describing problems, not designing solutions.
- Another issue is that structured decision-making may stifle innovation, especially if too much power is given to users over the technical leadership that has a vision for next-generation solutions.
- This pertains especially to the consulting domain, where the tendency is to "over-methodologize" everything, then apply a standard approach to all problems. This approach may result in poor and costly decisions. It was suggested that there be some way to allow disruptive thinking to occur in the Winbook iCBSP tool.

# 10 Capability and Gap Analysis of Commercial Tools

Requirements management in a net-centric environment is increasingly a problem encountered in industry domains. Commercial tools are starting to address these needs. To this end, we have conducted an analysis of the various capabilities offered, as well as the gaps not addressed, in three important categories identified by our extensive interaction with IT integration experts.

- Project planning – The capability to provide analysis tools supporting planning, management and progress reporting of requirements management in integration efforts.
- Data conversion – The capability to convert data (and metadata) in current/legacy systems to forms usable across a set of net-centric integrated systems.
- Knowledge management – The capability to capture and reuse knowledge about successful solutions to requirements management problems in integration contexts among net-centric partners, with the goal of developing a data repository of best-practices.

The following industry domains are considered and referenced in the tables that document the capabilities.

- M = Manufacturing
- P = Pharmaceuticals
- H = Healthcare
- AD = Aerospace & Defense
- R = Retail
- F = Finance
- G = Government
- NS = Non-specific

## 10.1 Project Planning Findings

Summary findings for project management tool capabilities and gaps include the following:

- Capabilities supported – traceability, scalability, requirements capture/analysis, file sharing, collaborative requirements management.
- Gaps – domain customization, architecture conflict and resolution, guidance for select vs. design, domain customization.

Table 8 and Table 9 document capabilities of specific tools that have been studied.

| | Main Industry Applications | Domain (Industry Application) Customization | Traceability | Align business goals with objectives | Capture and analyze requirements information | Task and/or milestone management | File sharing | Time tracking | Messaging System | To-do lists | Visibility |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Microsoft Project | NS | | X | X | | | X | X | X | | X |
| Basecamp | NS | | | | | X | X | X | X | X | |
| Wrike | NS | | | | | | X | X | X | | X |
| IBM RequisitePro | NS | | X | X | X | | X | | | | X |
| IBM Rational Doors | NS | | X | | X | | X | | | | X |
| IBM Requirement Composer | NS | | X | | X | X | X | | | | X |

**Table 8: Capabilities and Gaps of Project Planning Tools (1/2)**

| | Requirements-driven development process | Customizable dashboard | Collaborative Lifecycle Management - Requirements can be linked and traced end-to-end to test cases and development work items using lifecycle relationship links | Scalability to address requirements mgt needs | Manage changing requirements | Collaborative requirements mgt environment – allows stakeholders from different disciplines to contribute to the requirements process | Test tracking toolkit | Gap Analysis of unaddressed requirements | Guidance for Select vs. design – new decisions | IT architecture conflict identification & resolution |
|---|---|---|---|---|---|---|---|---|---|---|
| Microsoft Project | | X | | X | | | | | | |
| Basecamp | | | | | | | | | | |
| Wrike | | X | | | | | | | | |
| IBM RequisitePro | X | | | X | X | X | | | | |
| IBM Rational Doors | X | X | X | X | X | X | X | X | | |
| IBM Requirement Composer | X | X | X | X | | X | | X | | |

**Table 9: Capabilities and Gaps of Project Planning Tools (2/2)**

## 10.2 Data Conversion Findings

Summary findings for data conversion tool capabilities and gaps include the following:

- Capabilities supported – data transformation, data querying (extraction and analysis), data migration, data integration.
- Gaps – domain customization, data federation/aggregation from disparate sources, identification/reconciliation of data incompatibilities.

Table 10 and Table 11 document capabilities of specific tools that have been studied.

| | Track, Report and Share | Data Integration - combining data residing in different sources and providing users with unified view of the data | Cloud Integration - Integrating between on-premises and in cloud applications or databases | Data Migration - transferring data between storage types, formats, or computer systems | Data Synchronization | ETL (extraction, transformation, and loading) for Business Intelligence & Data Warehousing | Application integration to automate business processes | Versioning | Shared repository | Query - Extract & Analyze Data |
|---|---|---|---|---|---|---|---|---|---|---|
| SAP Legacy System Migration Workbench (LSMW) | | | | X | | | | | | X |
| SAP Business-Objects Data Integrator | | X | | | | X | | | | |
| WinShuttle Usability Platform | | | | | | | | | | X |
| MS Access | X | | | | | | | | X | X |
| Pervasive Data Integrator™ | X | X | X | X | | X | | | | X |
| Microsoft Application Platform - Enterprise Integration | | X | X | X | | | X | | | |
| TIBCO ActiveMatrix BusinessWorks | X | X | X | | | | | X | X | |
| Talend Integration Suite – Enterprise Edition | | X | | X | X | X | | X | X | X |
| Informatica Data Integration Product Suite | | X | X | X | X | X | | | | |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Oracle Data Integrator Enterprise Edition 11g | | X | | X | | X | | | | |
| Microsoft Amalga – formerly Azyxxi-Health Intelligence Integration Software | | X | | | | | | | X | X |

Table 10: Capabilities and Gaps of Data Conversion Tools (1/2)

| | Dashboard – activity monitoring console | SOA Manager / Webservices | Time-based scheduler | "Change data" capture | Data transformations – convert data between application-specific and common formats | Identification and/or reconciliation of data incompatibilities | Data federation - aggregate data from disparate sources into virtual databases | Main Industry Applications | Domain (Industry Application) Customization |
|---|---|---|---|---|---|---|---|---|---|
| SAP Legacy System Migration Workbench (LSMW) | | | | | | | | NS | |
| SAP Business-Objects Data Integrator | | | | | X | | X | NS | |
| WinShuttle Usability Platform | | | X | | | | | NS | |
| MS Access | | | | | | | | NS | |
| Pervasive Data Integrator™ | | | | | X | | | H, F NS | X |
| Microsoft Application Platform - Enterprise Integration | | | | | | | | NS | |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| TIBCO ActiveMatrix BusinessWorks | | | | | X | | | NS | |
| Talend Integration Suite – Enterprise Edition | X | X | X | X | X | X | | NS | |
| Informatica Data Integration Product Suite | | | | | | | X | NS | |
| Oracle Data Integrator Enterprise Edition 11g | | | | X | X | | | NS | |
| Microsoft Amalga – formerly Azyxxi-Health Intelligence Integration Software | | | | | | | | H | X |

**Table 11: Capabilities and Gaps of Data Conversion Tools (2/2)**

## 10.3 Knowledge Management Findings

Summary findings for knowledge management tool capabilities and gaps include the following:

- Capabilities supported – creation/publication/retrieval/sharing information, document search and classification, collaborative document work environments.
- Gaps – rationale capture, domain customization, taxonomies and classifications, integration of repositories.

Table 12 and Table 13 document capabilities of specific tools that have been studied.

| | Navigate through documents | Search and/or classify documents | Work with docs in a collaborative environment | Manage documents | Forums | Blogs | Wikis | Web 2.0 | Store Content |
|---|---|---|---|---|---|---|---|---|---|
| MS SharePoint | | X | X | X | | X | X | X | X |
| SAP Content Server | | | | | | | | | X |
| SAP Enterprise Knowledge Management | X | X | X | X | | | | | |
| SAP NetWeaver | | X | | | | | | | |
| IBM Connections | | | X | | X | X | X | | |

**Table 12: Capabilities and Gaps of Knowledge Management Tools (1/2)**

| | Create, publish, retrieve, & share information | Integration of repositories | Navigation in folders | Taxonomies & Classification | Content Management Services | Authoring & Publishing | Rationale Capture | Main Industry Applications | Domain (Industry Application) customization |
|---|---|---|---|---|---|---|---|---|---|
| MS SharePoint | X | | | | X | X | | NS | |
| SAP Content Server | | | | | | | | NS | |
| SAP Enterprise Knowledge Management | X | | | | | | | NS | |
| SAP NetWeaver | | X | X | X | X | X | | NS | |
| IBM Connections | | | | | | | | NS | |

**Table 13: Capabilities and Gaps of Knowledge Management Tools (2/2)**

# 11 CONCLUSION AND FUTURE WORK

## 11.1 CONCLUSION

This report has presented research results on how to manage capabilities and requirements across organizations that collaborate in a net-centric enterprise. The requirements management problem differs from that in single-system software development, since it occurs in the context of integrating and interoperating different systems across the various stakeholders. Thus, there is a strong analogy to systems of systems, and in particular acknowledged systems of systems.

We have presented a methodological framework for addressing the problem and have used a case study approach to determine important issues, relevance of our approach and effectiveness of various component MPTs within the approach. Along with this framework, we specified an integration taxonomy to provide a framework for documenting best (and less-than-successful) practices under different situations.

Within the methodological approach, two primary components are capabilities-to-requirements engineering and requirement-to-architectures engineering. The former adapts MPTs from systems of systems engineering to determine how best to decompose capabilities into requirements and assign responsibilities to constituent systems. The latter adapts MPTs used for traditional requirement-to-architecture efforts into an integration context for net-centric applicability and specifies a novel matrix approach to deriving integration architecture styles from desired integrated system properties.

Both these components are largely technical in nature, but have negotiation between stakeholders embedded. We received feedback from the user community that decision authority, negotiation and other socio issues are critical in successful efforts. To study effective practices and important issues in this regard more explicitly, we conducted a case study of the decision making processes behind how new capabilities were developed and implemented in systems that supported the system design and development phase of the F-35 Joint Strike Fighter. This SDD effort was conducted by a net-centric enterprise and was a transformative approach for the constituent organizations.

Finally, we conducted extensive validation of the concepts and MPTs developed in this research with subject matter experts in IT integration.

## 11.2 Future Research

There are a number of avenues of future research. Note that Appendix A provides direction for some of these items via user community feedback.

- Refinement of the integration taxonomy and development of additional case study examples that provide pointers to successful (unsuccessful) approaches to requirements management.
- Elaboration of the capabilities-to-requirements research with additional case study analysis and specification of negotiation methods recommended for system of system stakeholders.
- Elaboration of the requirements-to-architectures research by
    - Enhancing the social media implementation of iCBSP to determine which features promote effective negotiation and decision support,
    - Extending the integration matrix by (i) incorporating restrictions and limitations of the systems being integrated with respect to the architectural solutions that may be used, (ii) detecting mismatches that need to be resolved with guidelines for specific adapters/translators, and (iii) rank solutions based on prioritization of system properties, and
    - Determining constituent system attributes necessary to select the most appropriate integration solution.
- Leveraging of initial results of JSF SDD case study to study effective decision processes by (i) specifying formalized decision-making processes, decision rights, negotiation methods and tools, and knowledge capture methods and tools for lifecycle capability and requirements management, (ii) extending the case study to technical implementation of SDD capabilities, (iii) extending the case study to major partners in program enterprise, and/or (iv) extending the case study to downstream lifecycle phases (production/sustainment systems).
- Characterization of the complexity of the integration decision space to support schedule, cost, and risk assessments for different alternative decision prioritizations in the capabilities-to-architectures process and to gain insight into what types of prioritizations work best under what conditions.
- Specification and prototyping of a decision support environment that explicitly promotes knowledge management in net-centric requirements management, i.e., capturing insights and best practices, providing knowledge from past integration efforts to support current decisions.

## 11.3 Technology Transfer

Future work also encompasses technology transfer. One avenue for technology transfer is a set of training sessions or workshops at which research results are demonstrated and taught. Alternatives for a potential one-day seminar/workshop include the following.

1. Select a case study and provide a walk-through of how its requirements management would be addressed using the project's results, from identification of the type of system merger/integration, to identification and mitigation of socio decision issues, to capabilities-to-requirements decisions, to requirements-to-architectures decisions.
2. Present the methodology and associated MPTs, then demonstrate their use with the case study most appropriate to demonstrating the effectiveness of each one.
3. Select a case study and assign participants to roles (stakeholders, engineers, etc.). Engage in interactive use of selected parts of the methodology to demonstrate effectiveness. Other parts of the methodology may be presented as outcomes (i.e., not interactive). The particular assignment of interactive parts would need to be done keeping in mind time constraints.

# APPENDICES

## APPENDIX A: PRESENTATION FEEDBACK

This research was presented at the 2011 Annual SERC Research Review, and feedback was gathered from a set of attendees that included government and academic personnel. This feedback and associated responses are summarized in Table 14.

| Feedback | Follow-up |
|---|---|
| Socio issues in decision-making are important (e.g., decision-making authority, negotiation); technical issues were emphasized in the presentation. | • F-35 case study (current work)<br>• Winbook incorporation into iCBSP (current work)<br>• Socio and negotiation issues in SoS systems engineering (future work) |
| Cost is not considered. | Cost could be incorporated (future work) |
| Could properties in the integration matrix be prioritized or given weights? | Weights could be incorporated (future work) |
| Can relationships (e.g., correlations, conflicts) between integration matrix choices be represented? | Such relationships, especially lower level conflicts, could be identified using extension to COTS interoperability framework (future work) |
| There is interest in tools resulting from the project | Prototype tools are created to guide and support research issues; productization is left to others. |
| Validation of the integration matrix in various domains is not yet considered. For instance, certain domain characteristics may make particular architectural style elements more/less attractive. | Domain characteristics could be incorporated into existing matrix entries as explanations for a domain-specific matrix (current work). |

**Table 14: Research Feedback and Follow-up**

# Appendix B. Questionnaires for F-35 Joint Strike Fighter SDD Case Study

The F-35 Joint Strike Fighter SDD case study utilized two questionnaires to elicit information about socio decision-making in the net-centric enterprise that conducted the JSF program. The first surveyed the leadership of Lockheed Martin Aeronautics, which set the vision and framework for the SDD phase.

## B.1 Leadership

### B.1.1 Integrated Capability Context

1. What were the 3-5 most critical new enterprise-level capabilities required by the F-35 SDD contract in descending order of difficulty?
2. What existing capabilities could be leveraged to help meet these new capability intents? What were the gaps?
3. Was a change needed in how business is done? What needed to change?
4. What were the constraints?
5. Was the talent available to make the changes?
6. What was needed from leadership to drive the changes?
7. Did the required functional organizations have the resources/time to devote to the change?
8. Were the functional organizations aligned with, and supportive of, the different way of doing business (including Finance and HR)?
9. To what extent did the systems and practices used by the various stakeholder organizations support the new enterprise-level capabilities?
10. What was the specific intent for the "to-be" capabilities needed for the SDD phase of the F-35 Program, in terms of
    a. Managing the program,
    b. Capabilities to design and develop the F-35,
    c. Establishing the global team and concept of operations
    d. Creating the enabling skills, methods, tools and systems?
11. What was the outcome needed and when was it required?

### B.1.2 Transformation to Integrated Capabilities

1. How were the vision and the sense of urgency communicated to the global team, customers and other key stakeholders?
2. What were the major elements of the leadership strategy for achieving the SDD intents?
3. How was it assured that key stakeholders remain fully aligned with the intent and strategy for SDD?

4. What methods were used to establishing a "powerful coalition" between Program management Team (including partners, countries, etc), the Leadership Group, and customers that support "how the PM Team operates"?
5. How were conflicts identified and reconciled among the stakeholders? Were formal negotiation models used, or was the process ad hoc? In what areas did the process work vs. not work?
6. How was the collaboration/competition divergence addressed between LM Aero and partners in terms of knowledge/data sharing and system integration?
7. How was progress tracked on the overall integrated capability development or on a key capability(ies)?
   a. How would you ideally measure the integration progress?
8. To what extent did the functions and systems integrate successfully to support the capability intents? To what extent did legacy functions and systems impede progress? Were there other problems and constraints, and how were they addressed?
9. To what extent were there changes in capability intents or constraints during the integrated capability development process, and how were these handled?
   a. Internally imposed changes
   b. External imposed changes
10. What other obstacles not mentioned above arose during SDD, and how did the leadership deal with them?
11. Was the F-35 SDD successful in terms of SDD intent and foundational support for the production, test and sustainment phases of the Program?
12. Were knowledge capture plans used to create a repeatable integrated capability transformation process for other programs?
13. To what extent was the change anchored in the "corporate culture," and how was this done?
14. What would you have done differently in hindsight?

## B.2 Technical Management

### B.2.1 Integrated Capability Context – F-35 Global Digital Lifecycle Design

1. What existing capabilities within your organization could be leveraged to help meet the global digital design capability intent?
2. What new capabilities were needed?
3. What capabilities within other organizations (LM Aero, partners, and suppliers) existed that could be leveraged to help meet the digital design capability intent?
4. What new capabilities were needed from them?
5. To what extent did the systems and practices used within your organizations support the global digital design capability?
6. What changes were needed within your organization in how business was done?
7. What were the constraints?
8. Was the talent available to make the changes?

9. Did your organization and others have the resources/time to devote to the F-35 SDD transformative change?
10. Were the participating organizations aligned with, and supportive of, the different way of doing business (including Finance and HR)?
11. What was needed from the LM Aero leadership group (Program, Functions and Company) to drive the changes across the enterprise?
12. How did your organization support other enterprise-level integrated capability intents for SDD, and how did this influence your ability to support global digital design capability?
13. What was the specific intent for the "to-be" capabilities needed from your organization for the SDD phase of the F-35 Program, in terms of
    a. Helping manage the program,
    b. Capabilities to design and develop the F-35,
    c. Helping establishing the global team and concept of operations
    d. Creating the enabling skills, methods, tools and systems?
14. What was the outcome needed from your organization and others, and when was it required?

### B.2.2 Transformation to Integrated Capabilities – Global Digital Design Strategy

1. What was the inter-organizational strategy to achieve the integrated digital design capability, and how was it arrived at? How did your organization team with others (LM Aero, partners, and major suppliers) to realize the strategy?
2. Did your organization understand the vision and the sense of urgency?
3. How did your organization support the major elements for achieving the SDD intents?
4. How did your organization work to remain fully aligned with the intent and strategy for SDD?
5. How were conflicts identified and reconciled among the various collaborating organizations? Were formal negotiation models used, or was the process ad hoc? In what areas did the process work vs. not work?
6. How did the collaboration/competition divergence between LM Aero and partners in terms of knowledge/data sharing and system integration affect the global digital design capability development and your organization's efforts?
7. How did you report progress from your organization on the overall integrated digital design capability development, and how was this combined with progress of collaborating organizations?
    a. How would you ideally measure the integration progress?
8. To what extent were there changes in capability intents or constraints during the integration process, and how were these handled?
    a. Internally imposed changes
    b. External imposed changes
9. What other obstacles not mentioned above arose during SDD, and how did your organization deal with them?

10. Was the F-35 SDD successful in terms of SDD intent and foundational support for production, test and sustainment phases of the Program?
11. Were knowledge capture plans used in your organization to create a repeatable integrated capability transformation process for other programs?
12. To what extent was the change anchored in the "corporate culture"?
13. What would your organization have done differently in hindsight?

### B.2.3 Transformation to Integrated Capabilities – Global Digital Design Implementation

1. How did the process of decomposing the overall global integrated digital design capability into intermediate-level capabilities then into requirements work? Was there iteration? Would this process (or parts of it) be repeatable?
2. To what extent were rationales for intermediate-level capabilities captured?
3. How were tasks prioritized within your organization in the integrated digital design capability development?
4. What types of information did your organization and others share between each other? For example, did they share design rules, interfaces, metadata, high-level capabilities, internal requirements repositories, access to the internal systems, etc.?
5. How many distinct systems needed to be integrated? What were they?
6. How dynamic was the integration? Were there other integrations involving the same systems that started afterward or that were ongoing in parallel?
7. How did the existing platforms and technologies (e.g., operating system, middleware, and programming languages) affect the integration process and integration decisions?
8. Was sufficient technical documentation for all involved systems available?
    a. Was any technical information unavailable due to security, intellectual property, or other concerns?
9. Were there constraints regarding availability of the constituent systems to their users while the systems were under integration?
10. Was there system downtime planned to support migration to system upgrades? If so, how frequent, extensive were these?
11. Were there plans to unwind the integration?
    a. Points at which the contract/program might not have moved forward?
    b. Contingency plans if the intent was to have the integration last over the lifecycle?
12. Did your organization use any decision tools to support the integration? What type of tools would you have liked to use?
13. What would your organization have done differently in hindsight?

## APPENDIX C: REFERENCES

ANSI/IEEE. (2006). ANSI/IEEE Standard 1471: Recommended Practice for Architectural Description of Software-Intensive Systems Retrieved April 1, 2011, from http://www.iso-architecture.org/ieee-1471/

Basole, R. C., & DeMillo, R. A. (2006). Enterprise IT and Transformation. In W. B. Rouse (Ed.), *Enterprise Transformation: Understanding and Enabling Fundamental Change* (pp. 223-252). Hoboken, NJ: Wiley-Interscience.

Bennett, N., Kessler, W. C., & McGinnis, L. F. (Eds.). (in press). *Enterprise Transformation: Manufacturing in a Global Enterprise*: IOS Press.

Bodner, D. A., Medvidovic, N., Rouse, W. B., Boehm, B. W., DeMillo, R. A., Edwards, G., . . . Pradhan, A. (2011). RT-25: Requirements Management for Net-Centric Enterprises: Technical Report SERC-2011-TR-017, Systems Engineering Research Center, Hoboken, NJ.

Boehm, B., Grunbacher, P., & Briggs, R. O. (2001). Developing Groupware for Requirements Negotiation: Lessons Learned. *IEEE Software*, 46-55.

Boehm, B., Valerdi, R., Lane, J. A., & Brown, W. (2005). COCOMO Suite Methodology and Evolution. *CrossTalk Journal, 18*(4), 20-25.

Burgelman, R. A., & Meza, P. E. (2004). HP and Compaq Combined: In Search of Scale and Scope. Stanford, CA: Graduate School of Business, Stanford University.

CJCS. (2007). Operation of the Joint Capabilities Integration and Development System. Washington, DC: Chairman of the Joint Chiefs of Staff, Department of Defense.

DoD. (1998). *C4ISR Architecture Working Group Final Report - Levels of Information System Interoperability (LISI)*. Washington, DC: OSD(ASD(C31)) C4ISR AWG.

DoD. (2008). *Systems Engineering Guide for Systems-of-Systems, Version 1.0*. Washington, DC: OUSD(A&T)SSE.

Fielding, R. T., & Taylor, R. N. (2002). Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology, 2*(2), 115-150.

Greenwood, D., & Sommerville, I. (2011). *Responsibility Modeling for Identifying Sociotechnical Threats to the Dependability of Coalitions of Systems*. Paper presented at the 2011 6th International Conference on Systems of Systems Engineering, Albuquerque, NM.

Grünbacher, P., Egyed, A., & Medvidovic, N. (2004). Reconciling Software Requirements and Architectures with Intermediate Models. *Software and Systems Modeling, 3*(3), 235-253.

Land, R., & Crnkovic, I. (2011). Oh Dear We Bought Our Competitor: Integrating Similar Software Systems. *IEEE Software, 28*(2), 75-82.

Lane, J. A. (2009). *Cost Model Extensions to Support Systems Engineering Cost Estimation for Complex Systems and Systems of Systems*. Paper presented at the 7th Annual Conference on Systems Engineering Research, Loughborough University, UK.

Lane, J. A., & Bohn, T. (2010). Using SysML to Evolve Systems of Systems: Technical Report USC-CSSE-2010-506, Center for Systems and Software Engineering, University of Southern California, Los Angeles, CA.
Rouse, W. B. (Ed.). (2006). *Enterprise Transformation: Understanding and Enabling Fundamental Change*. Hoboken, NJ: Wiley-Interscience.
Solano, M. A. (2011). *SoSE Architecture Principles for Net-Centric Multi-Int Fusion Systems*. Paper presented at the 2011 6th International Conference on Systems of Systems Engineering, Albuquerque, NM.
USAF-SAB. (2005). *Report on System-of-Systems Engineering for Air Force Capability Development*. (Public Release SAB-TR-05-04). Washington, DC: United States Air Force (USAF) Scientific Advisory Board (SAB).
Wu, D., Yang, D., & Boehm, B. W. (2009). Finding Success in Rapid Collaborative Requirements Negotiatioon Using Wiki and Shaper: Technical report TR 2009-519, Center for Systems and Software Engineering, University of Southern California.